

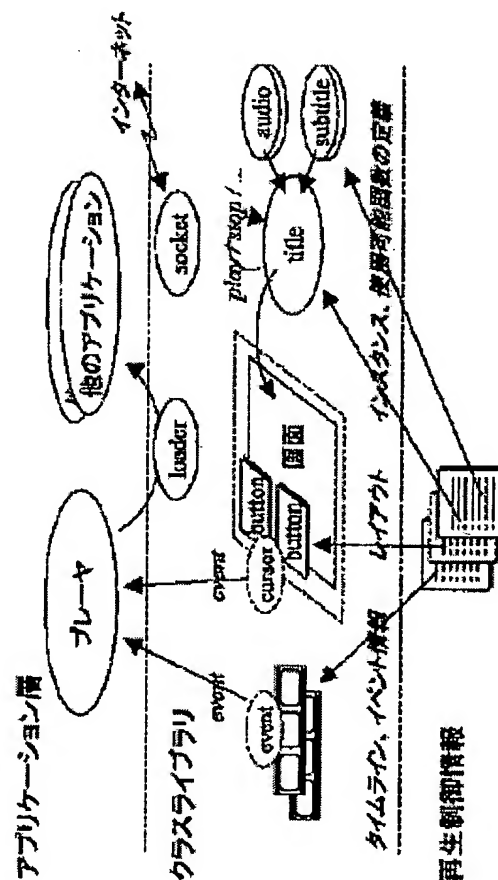
IMAGE PLAYBACK EQUIPMENT FOR DIGITAL IMAGE CONTENTS, IMAGE PLAYBACK METHOD, IMAGE PLAYBACK PROGRAM, AND PACKAGE MEDIA

Patent number: JP2002369154
Publication date: 2002-12-20
Inventor: OKADA TOMOYUKI; IKEDA KO; NAKAMURA KAZUHIKO
Applicant: MATSUSHITA ELECTRIC IND CO LTD
Classification:
 - international: **H04N5/76; G11B20/10; H04N5/93; H04N5/76; G11B20/10; H04N5/93; (IPC1-7): H04N5/93; G11B20/10; H04N5/76**
 - european:
Application number: JP20020100118 20020402
Priority number(s): JP20020100118 20020402; JP20010103375 20010402

Report a data error here

Abstract of JP2002369154

PROBLEM TO BE SOLVED: To provide a technique for putting adding new value to contents and a technique for managing the added value of contents. **SOLUTION:** This image playback equipment plays back package media. This package media include image contents wherein image data and playback control information are stored, and an extended application to use the image contents. This image playback equipment includes an OS, middleware which absorbs the difference of function corresponding to the kind of that OS, and a player application which operates on the middleware and plays back the image contents, as software which is preserved in an inner memory prior to execution. The middleware has a class library including a tool to playback the player application and a tool for execution. The player application plays back the image contents via the above tool. The extended application is executed via the above tool.



Data supplied from the esp@cenet database - Worldwide

【特許請求の範囲】

【請求項1】 外部から供給されるパッケージメディアを再生する映像再生装置であって、

前記パッケージメディアは、

映像データと、前記映像データの再生を制御する再生制御情報とを所定のデータフォーマットで格納する映像コンテンツと、

前記映像コンテンツを使用する拡張アプリケーションソフトウェアとを含み、

前記映像再生装置は、

内部メモリに予め保持され実行されるソフトウェアとして、

複数の種類のオペレーティングシステムから選ばれた一つのオペレーティングシステムと、

前記オペレーティングシステムの種類に対応した機能の差異を吸収するミドルウェアと、

前記ミドルウェア上で動作し、前記映像コンテンツを再生するプレーヤアプリケーションソフトウェアとを備え、

前記ミドルウェアは、

前記プレーヤアプリケーションソフトウェアが前記パッケージメディアを再生したり前記拡張アプリケーションソフトウェアを実行するために用いるツールを含むクラスライブラリを有し、

前記プレーヤアプリケーションソフトウェアは、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して、前記パッケージメディアの前記映像コンテンツを、前記所定のフォーマットに従って統一的に再生し、

前記拡張アプリケーションソフトウェアは、同一のパッケージメディアに含まれる前記映像コンテンツを用いて、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して実行される、映像再生装置。

【請求項2】 前記映像再生装置は、再生ステータス情報を管理しており、前記パッケージメディアの前記再生制御情報は前記再生ステータス情報に対応する再生制限情報を含み、前記再生制御情報を解析し、前記再生制御情報内の再生制限情報と前記再生ステータス情報の比較により、前記拡張アプリケーションソフトウェアが前記ミドルウェアの前記クラスライブラリに含まれる前記ツールのいずれかを無効に設定する、請求項1に記載の映像再生装置。

【請求項3】 外部から供給されるパッケージメディアを映像再生装置で再生する映像再生方法であって、

前記パッケージメディアは、

映像データと、前記映像データの再生を制御する再生制御情報とを所定のデータフォーマットで格納する映像コンテンツと、

前記映像コンテンツを使用する拡張アプリケーションソフトウェアとを含み、

前記映像再生方法は、

複数の種類のオペレーティングシステムから選ばれた一つのオペレーティングシステムを前記映像再生装置の内部メモリに読み込み、起動するステップと、

該オペレーティングシステムの種類に対応した機能の差異を吸収するミドルウェアであって、該ミドルウェア上で動作するアプリケーションソフトウェアが前記パッケージメディアを再生したり、実行するために用いるツールを含むクラスライブラリを有するミドルウェアを前記映像再生装置の前記内部メモリに読み込み、起動するステップと、

前記ミドルウェア上で動作し、前記映像コンテンツを再生するプレーヤアプリケーションソフトウェアを前記映像再生装置の内部メモリに読み込み、起動するステップと、

前記ミドルウェア上で動作し、前記映像コンテンツを使用する拡張アプリケーションソフトウェアを前記映像再生装置の内部メモリに読み込み、起動するステップと、前記プレーヤアプリケーションソフトウェアによって、前記ミドルウェアの前記クラスライブラリに含まれるツールを介して、前記パッケージメディアの前記映像コンテンツを、前記所定のフォーマットに従って統一的に再生するステップと、

前記拡張アプリケーションソフトウェアによって、前記映像コンテンツを用いて、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して実行するステップとを含む映像再生方法。

【請求項4】 外部から供給されるパッケージメディアを再生する映像再生プログラムであって、

前記パッケージメディアは、

映像データと、前記映像データの再生を制御する再生制御情報とを所定のデータフォーマットで格納する映像コンテンツと、

前記映像コンテンツを使用する拡張アプリケーションソフトウェアとを含み、

前記映像再生プログラムは、

内部メモリに予め保持され実行されるソフトウェアとして、

複数の種類のオペレーティングシステムから選ばれた一つのオペレーティングシステムと、

前記オペレーティングシステムの種類に対応した機能の差異を吸収するミドルウェアと、

前記ミドルウェア上で動作し、前記映像コンテンツを再生するプレーヤアプリケーションソフトウェアとを備え、

前記ミドルウェアは、

前記プレーヤアプリケーションソフトウェアが前記パッケージメディアを再生したり前記拡張アプリケーションソフトウェアを実行するために用いるツールを含むクラスライブラリを有し、

前記プレーヤアプリケーションソフトウェアは、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して、前記パッケージメディアの前記映像コンテンツを、前記所定のフォーマットに従って統一的に再生し、

前記拡張アプリケーションソフトウェアは、同一のパッケージメディアに含まれる前記映像コンテンツを用いて、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して実行される、映像再生プログラム。

【請求項5】 請求項4に記載の前記映像再生プログラムを格納するコンピュータ読み取り可能な記録媒体。

【請求項6】 映像再生装置に外部から供給され、該映像再生装置で再生されるパッケージメディアであって、前記パッケージメディアは、映像データと、前記映像データの再生を制御する再生制御情報とを所定のデータフォーマットで格納する映像コンテンツと、

前記映像コンテンツを使用する拡張アプリケーションソフトウェアとを含み、

前記映像再生装置は、内部メモリに予め保持され実行されるソフトウェアとして、

複数の種類のオペレーティングシステムから選ばれた一つのオペレーティングシステムと、

前記オペレーティングシステムの種類に対応した機能の差異を吸収するミドルウェアと、

前記ミドルウェア上で動作し、前記映像コンテンツを再生するプレーヤアプリケーションソフトウェアとを備え、

前記ミドルウェアは、

前記プレーヤアプリケーションソフトウェアが前記パッケージメディアを再生したり前記拡張アプリケーションソフトウェアを実行するために用いるツールを含むクラスライブラリを有し、

前記プレーヤアプリケーションソフトウェアは、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して、前記パッケージメディアの前記映像コンテンツを、前記所定のフォーマットに従って統一的に再生し、

前記拡張アプリケーションソフトウェアは、同一のパッケージメディアに含まれる前記映像コンテンツを用いて、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して実行される、パッケージメディア。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、映画などのデジタル映像コンテンツを記録したパッケージメディアと、その映像再生装置、映像再生方法、及び映像再生プログラ

ムに関する。特に、DVDを置きかえる技術であるEパッケージに関する。

【0002】

【従来の技術】 (パッケージビジネス) 最初にパッケージビジネスの推移について説明する。図1は、現在および今後のパッケージビジネスの流通形態を示している。図1に示すように、パッケージビジネスの流通とは、コンテンツプロバイダが所持する映画などのコンテンツを、いかにしてユーザへ提供するかである。

【0003】 近年、映画などのコンテンツは、DVDによって、コンテンツプロバイダからユーザに提供される。旧来のビデオカセットによる流通に比べて、DVDではスタンパによる製造コストの削減、小スペースによる輸送コストの削減、店舗における展示スペースの削減など流通ビジネスの効率を大きく向上させることができた。DVDには、高画質、高音質、ランダムアクセス機能、マルチアングル等のインタラクティブ機能などの付加価値が加わり、ビデオカセットに比べ大きなメリットがある。

【0004】 次にコンテンツの価値について説明する。図2は、コンテンツの価値を示す概念図である。従来からのビデオカセットは、テープ上にリニアにタイトルを記録していた。即ち、映画館での上映と同じく、映画そのものを供給する媒体であり、それ以上の価値を有していなかった。DVDは、映画自身が持つ価値の他にも、マルチアングルやマルチストーリーといったインタラクティブ機能や、メニューからのタイトル選択とランダムアクセス、音声や字幕のマルチリンガル対応などの付加価値を有する。

【0005】 コンテンツの価値は、様々な要因によって目減りをしていく。例えば、音楽には「流行」がある。音楽コンテンツの多くは、時間と共に、流行が変わりその価値は著しく目減りする。映画もまた同様にその傾向がある。一方で、映画には「ストーリー」が含まれている。話しの展開を知りたい視聴者は続きを見る。逆に、ストーリーを知ってしまった視聴者にとって、続きを見る動機は薄れてしまう。即ち、個人にとって、コンテンツの価値が目減りする。同じ音楽を毎日聴く人が多いのに対して、同じ映画を毎日見る人が少ないのは上述の為である。統計的に考えれば、映画を見た人が増えるに従って、そのコンテンツの持つ市場での価値は徐々に減っていく。

【0006】 図3は、時間軸におけるコンテンツの価値と、それに合わせた映画ビジネスを示した図である。横軸が時間を示し、縦軸がコンテンツの価値を示している。映画では、タイムシフトと呼ばれる独特のビジネスモデルを持っている。映画は最初に映画館で上映され、その後、DVDなどのパッケージソフトとして個人向けに販売が行われる。続いて、衛星放送やケーブルテレビなどを使って、Pay Per Viewと呼ばれる有

料視聴に供せられ、最後に地上波局による無料放送に供せられる。地上波放送では、もちろん個々人は無料視聴できるが、これは、企業スポンサーによる広告収入によって支えられている。

【0007】(DVDの例)ここで、DVDを例に従来のパッケージビジネスを支える技術を説明する。特に断りが無い限り本文で言うDVDはDVD-ROM、即ち再生専用ディスクを指し、DVD-RAM等の録再可能なディスクを指してはいない。

【0008】図4は、DVDに記録されているデータの構造を示す。DVDディスクには、約4.7GB(ギガバイト)の記録領域があり、DVDドライブのサーボを安定回転させるリードインと呼ばれる領域に続いて、「0」または「1」の2値を記録する論理アドレス空間が続く、最後にディスク記録領域の終了を示すリードアウト領域が続いている。

【0009】論理アドレス空間には、まずファイルシステム領域があり、続いてAVデータや映画のシナリオを記述したナビゲーションデータが記録されている。ファイルシステムとは、データをファイルとディレクトリ(フォルダ)として管理するシステムであり、DVDディスク上に記録されているAVデータやナビゲーションデータは全てファイルシステムを通してそれぞれのディレクトリ、ファイルとして扱うことができる。

【0010】図4に示すように、DVDディスク上では、「VIDEO_TS」と呼ばれる、DVDビデオタイトルを格納するディレクトリがルートディレクトリ直下に置かれている。このディレクトリの中には、シナリオ管理やインタラクティブ性を実現するナビゲーション情報を記録した「VIDEO_TS.IFO」、「VTS_01_0.IFO」などのファイルと、AVデータを記録した「VTS_01_0.VOB」のファイルとが置かれている。

【0011】AVデータとしては、ISO/IEC13818(MPEG)に準拠したストリームが記録されている。DVDでは、MPEGストリーム1本はVOBとよばれ、複数のVOBが拡張子「.VOB」を有するファイルに記録されている。複数のVOBが順に1つのVOBファイルに記録されていくが、VOBファイルが1GBを超えた場合には、1GBを境界に複数のVOBファイルに分割されて記録されていく。

【0012】ナビゲーション情報は、「VMGI」と呼ばれるディスク全体を管理する情報と、「VTSI」と呼ばれるタイトル個々に関する情報とに大別される。「VTSI」の中には、「VOB(MPEGストリーム)」の一部または、全部を一つの再生ユニットとする「Cell」を有する「PGC情報」が含まれる。「Cell」は、再生シーケンスを定義する。ここで重要なのは、「Cell」が「VOB」の一部または全部を示すために用いているのが、論理アドレス空間を基準とし

たアドレス情報であることである。

【0013】例えば、コンピュータ上のHDD(ハードディスクドライブ)の場合は、ファイルが記録、編集、消去が繰り返し行われるため、同じファイルであっても、HDD上では常に同じ場所に記録されている保証はない。ファイルシステムの最大の特徴は、ファイルがHDD上のどこに記録されていてもアプリケーションからは同じようにファイルが扱えることである。

【0014】DVDは、AVとPCの融合を図るため、ファイルシステムを載せながらも、論理アドレスを意識したデータ構造を取っている。民生用AV機器の性能はPCに遠く及ばない。特に、DVD登場の当時の性能はファイルシステムを載せることすら危ぶまれていた。しかし、DVDは民生機器だけでなく、PCでの使用も大きく期待されていた。事実、最近のPC製品でDVD再生機能搭載機が珍しくない。

【0015】即ち、DVDは、民生機としての現実的な性能と、PCからのアクセスとの両方が望まれていた。このため、PCではファイルシステムを通してデータにアクセスでき、一方、ファイルシステムの機能を持たない民生用AV機器では、論理アドレスベースでデータにアクセスが出来るように、DVDは設計された。こうして、DVDは民生機器、PC双方の幅広い支持を得ることが可能になった。

【0016】

【発明が解決しようとする課題】(DVD規格の課題)図1で現在および今後のパッケージビジネスの流通形態を説明している。図1に示すように、インターネットの爆発的な普及や、デジタル放送の実用化によって、パッケージの流通は物理ディスクを用いた方法だけでなくなっている。

【0017】既に、幾つかのコンテンツは、インターネットでストリーム配信されている。また、近年、一時記憶媒体としてハードディスクドライブ(HDD)を搭載したセットトップボックス(STB)が現れている。このハードディスクドライブに、デジタル放送を蓄積し、後から視聴できる。このように、コンテンツビジネスを行う上での環境は劇的に変化している。

【0018】また、映画コンテンツの流通も、DVD等の物理媒体による流通から、デジタル放送波やインターネットを使用した電子流通へとシフトしていくと考えられる。

【0019】図5は家庭内におけるAV機器の構成を示す図である。AV機器を取り囲む環境は、インターネット、デジタル放送によって大きく変わっている。例えば、AV機器のインターネット接続や、デジタル放送を受信するセットトップボックス(STB)とレコーダやテレビとの接続等、機器間での接続、即ち、家庭内ネットワークが必須になっている。

【0020】特にデジタル放送によるコンテンツ配信

は、インターネットのようなユーザからの要求に応じるプル型ではなく、一方的にデータを送りつけるプッシュ型である。即ち、一方的に全ての視聴者に対してコンテンツを送り届ける。この場合コンテンツの著作権を防ぐシステムが必要になる。著作権保護のシステムは、暗号化技術やそのシステム技術であるDRM（デジタル・ライト・マネージメント）によって実用化されつつある。

【0021】次に必要となる技術は、コンテンツの持つ価値を管理する技術である。例えば、図2に示したように、既存のDVDより更に進んだ付加価値と、図3で示すタイムシフトモデルのように、時期や流通状況に合わせたコンテンツの価値管理方式が必要になってくる。既存のDVDは、ディスクの売り切り（セルスルー）を基本としているため、新たな付加価値の追加や、管理などが出来る構造になってはいない。

【0022】（コンテンツ流通の課題）コンテンツ流通の課題として、デジタル放送システムの乱立が挙げられる。日本国内では、CSデジタル放送、BSデジタル放送が実用化されており、更にCS110°と呼ばれる新たな衛星放送と地上波デジタル放送とが始まろうとしている。また、欧州では、各国毎に異なるデジタル放送システムがあるが、各国のシステムはDVB（Digital Video Broadcasting）システムとして統一方向に向かっている。ただし、このDVBシステムは日本のシステムとは異なる。北米でもATISと呼ばれる独自方式が検討されている。

【0023】デジタル放送では、現在のアナログ放送のNTSC、PAL方式等よりも更に複雑に地域毎にシステムが異なってくる。このため、映画のような全世界を対象としたコンテンツの場合、地域毎でのオーサリングが必要となり、製作コストの高騰が予想される。

【0024】その一つの解決方法として、DVDと同等のコンテンツを電子配信する全世界統一の電子配信用パッケージが考えられる。しかし、この電子配信用パッケージをそのまま有料放送や無料地上波放送と置き換えてしまうと、DVDと全く同等のコンテンツを無料地上波放送でも楽しめるため、ユーザのDVD購買欲を低下させ、DVDビジネスを崩壊させる危険性がある。

【0025】そのため、コンテンツの配信時期に応じて新たな付加価値をつける技術と、例えば、ユーザに応じてコンテンツの再生使用制限を設ける等の付加価値を管理する技術とを必要としている。

【0026】そこで、本発明の目的は、上述の課題「コンテンツに付加価値をつける」、「時期や流通形態に合わせたコンテンツの価値管理」を解決することである。具体的には、ネットワーク時代に対応した新しいコンテンツビジネスを築く技術であるEパッケージを提供することである。

【0027】

【課題を解決するための手段】本発明に係る映像再生装置は、外部から供給されるパッケージメディアを再生する映像再生装置であって、前記パッケージメディアは、映像データと、前記映像データの再生を制御する再生制御情報とを所定のデータフォーマットで格納する映像コンテンツと、前記映像コンテンツを使用する拡張アプリケーションソフトウェアとを含み、前記映像再生装置は、内部メモリに予め保持され実行されるソフトウェアとして、複数の種類のオペレーティングシステムから選ばれた一つのオペレーティングシステムと、前記オペレーティングシステムの種類に対応した機能の差異を吸収するミドルウェアと、前記ミドルウェア上で動作し、前記映像コンテンツを再生するプレーヤアプリケーションソフトウェアとを備え、前記ミドルウェアは、前記プレーヤアプリケーションソフトウェアが前記パッケージメディアを再生したり前記拡張アプリケーションソフトウェアを実行するために用いるツールを含むクラスライブラリを有し、前記プレーヤアプリケーションソフトウェアは、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して、前記パッケージメディアの前記映像コンテンツを、前記所定のフォーマットに従って統一的に再生し、前記拡張アプリケーションソフトウェアは、同一のパッケージメディアに含まれる前記映像コンテンツを用いて、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して実行される。

【0028】この映像再生装置は、Eパッケージの映像コンテンツを再生する。上記オペレーティングシステムは、例えば、マイクロソフト社によるウィンドウズ（登録商標）、アップル社によるMac OS（登録商標）、又はフリーウェアのLinux等を用いることができる。なお、オペレーティングシステムは上記のものに限られず、各メーカーによるOSを含む。また、上記ミドルウェアとしては、例えば、Java（登録商標）を用いることができる。このミドルウェアによってオペレーティングシステムの種類が異なる場合にも機能の差異を吸収できる。また、上記プレーヤアプリケーションソフトウェアは、パッケージメディアの映像コンテンツを再生する。さらに、拡張アプリケーションソフトウェアとしては、例えば、上記パッケージメディアの映像コンテンツを利用するゲームアプリケーション等である。なお、上記プレーヤアプリケーションソフトウェア及び拡張アプリケーションソフトウェアは、上記ミドルウェア上で動作する。ミドルウェアは、上記アプリケーションソフトウェアが映像コンテンツを再生したり、実行する際に使用されるツールを含むクラスライブラリを有する。このクラスライブラリに含まれるツールとは、例えば、種々の機能を実現するクラスやそのメンバ関数を意味する。また、さらに、この映像再生システムは、ネットワークを介して配信されたソフトウェアを実行することによって構成することもできる。

【0029】また、この映像再生装置は、再生ステータス情報を管理しており、前記パッケージメディアの前記再生制御情報は前記再生ステータス情報に対応する再生制限情報を含み、前記再生制御情報を解析し、前記再生制御情報内の再生制限情報と前記再生ステータス情報の比較により、前記拡張アプリケーションソフトウェアが前記ミドルウェアの前記クラスライブラリに含まれる前記ツールのいずれかを無効に設定する。

【0030】本発明に係る映像再生方法は、外部から供給されるパッケージメディアを映像再生装置で再生する映像再生方法であって、前記パッケージメディアは、映像データと、前記映像データの再生を制御する再生制御情報とを所定のデータフォーマットで格納する映像コンテンツと、前記映像コンテンツを使用する拡張アプリケーションソフトウェアとを含み、前記映像再生方法は、複数の種類のオペレーティングシステムから選ばれた一つのオペレーティングシステムを前記映像再生装置の内部メモリに読み込み、起動するステップと、該オペレーティングシステムの種類に対応した機能の差異を吸収するミドルウェアであって、該ミドルウェア上で動作するアプリケーションソフトウェアが前記パッケージメディアを再生したり、実行するために用いるツールを含むクラスライブラリを有するミドルウェアを前記映像再生装置の前記内部メモリに読み込み、起動するステップと、前記ミドルウェア上で動作し、前記映像コンテンツを再生するプレーヤアプリケーションソフトウェアを前記映像再生装置の内部メモリに読み込み、起動するステップと、前記ミドルウェア上で動作し、前記映像コンテンツを使用する拡張アプリケーションソフトウェアを前記映像再生装置の内部メモリに読み込み、起動するステップと、前記プレーヤアプリケーションソフトウェアによって、前記ミドルウェアの前記クラスライブラリに含まれるツールを介して、前記パッケージメディアの前記映像コンテンツを、前記所定のフォーマットに従って統一的に再生するステップと、前記拡張アプリケーションソフトウェアによって、前記映像コンテンツを用いて、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して実行するステップとを含む。

【0031】本発明に係る映像再生プログラムは、外部から供給されるパッケージメディアを再生する映像再生プログラムであって、前記パッケージメディアは、映像データと、前記映像データの再生を制御する再生制御情報とを所定のデータフォーマットで格納する映像コンテンツと、前記映像コンテンツを使用する拡張アプリケーションソフトウェアとを含み、前記映像再生プログラムは、内部メモリに予め保持され実行されるソフトウェアとして、複数の種類のオペレーティングシステムから選ばれた一つのオペレーティングシステムと、前記オペレーティングシステムの種類に対応した機能の差異を吸収するミドルウェアと、前記ミドルウェア上で動作し、前

記映像コンテンツを再生するプレーヤアプリケーションソフトウェアとを備え、前記ミドルウェアは、前記プレーヤアプリケーションソフトウェアが前記パッケージメディアを再生したり前記拡張アプリケーションソフトウェアを実行するために用いるツールを含むクラスライブラリを有し、前記プレーヤアプリケーションソフトウェアは、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して、前記パッケージメディアの前記映像コンテンツを、前記所定のフォーマットに従って統一的に再生し、前記拡張アプリケーションソフトウェアは、同一のパッケージメディアに含まれる前記映像コンテンツを用いて、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して実行される。

【0032】また、本発明に係るコンピュータ読み取り可能な記録媒体は、前記映像再生プログラムを格納する。

【0033】本発明に係るパッケージメディアは、映像再生装置に外部から供給され、該映像再生装置で再生されるパッケージメディアであって、前記パッケージメディアは、映像データと、前記映像データの再生を制御する再生制御情報とを所定のデータフォーマットで格納する映像コンテンツと、前記映像コンテンツを使用する拡張アプリケーションソフトウェアとを含み、前記映像再生装置は、内部メモリに予め保持され実行されるソフトウェアとして、複数の種類のオペレーティングシステムから選ばれた一つのオペレーティングシステムと、前記オペレーティングシステムの種類に対応した機能の差異を吸収するミドルウェアと、前記ミドルウェア上で動作し、前記映像コンテンツを再生するプレーヤアプリケーションソフトウェアとを備え、前記ミドルウェアは、前記プレーヤアプリケーションソフトウェアが前記パッケージメディアを再生したり前記拡張アプリケーションソフトウェアを実行するために用いるツールを含むクラスライブラリを有し、前記プレーヤアプリケーションソフトウェアは、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して、前記パッケージメディアの前記映像コンテンツを、前記所定のフォーマットに従って統一的に再生し、前記拡張アプリケーションソフトウェアは、同一のパッケージメディアに含まれる前記映像コンテンツを用いて、前記ミドルウェアの前記クラスライブラリに含まれる前記ツールを介して実行される。

【0034】このパッケージメディアは、高付加価値を有するEパッケージである。即ち、このパッケージメディアの映像コンテンツをプレーヤアプリケーションソフトウェアで再生するだけでなく、上記映像コンテンツを利用するゲームアプリケーションソフトウェア等を関連させて実行することができる。また、このパッケージメディアでは、再生制御情報には、映像データの再生順序を規定するシナリオ情報を含んでいてもよい。さらに、

再生制御情報は、映像コンテンツの再生やゲームアプリケーションの利用を制限するレベルを規定する再生レベル情報を含んでいてもよい。

【0035】

【発明の実施の形態】以下、図面を参照して本発明に係る好ましい実施形態について説明する。なお、図面において、同一の符号は同様のものを示す。

【0036】(新たなビジネスの形態)図2、図3で説明したように、映画コンテンツの価値は時間と共に減少をしていく。更に、世界的に電子流通を進めていくためには、既存のビジネスモデルを変えていかなければならない。

【0037】本発明の実施の形態に係るデジタル映像コンテンツを含むパッケージメディア(以下、「Eパッケージ」という)は、図2で示すように映画そのものに匹敵するアプリケーションを付加価値として導入する。これにより、パッケージの価値を高めることができる。また、図5で示すように、アプリケーションによって導かれる価値をコントロールし、パッケージにレベルを持たせ、同じタイトルであっても差別化が図れるようにする。

【0038】例えば、図8で示しているように、全てのアプリケーションを使用可能としたパッケージを「フルパッケージ」、一部のアプリケーションの使用を制限したパッケージを「制限付きパッケージ」、映画コンテンツだけの視聴を可能としたパッケージを「無料パッケージ」としてレベルを設けて価値をコントロールする。

【0039】図3で示しているパッケージビジネスにおいて、既存のDVDに代わってフルパッケージを流通させ、有料放送に代わって制限付きパッケージを、無料放送に代わって無料パッケージをそれぞれ流通させることができる。本実施の形態では、3通りのレベルでしか説明を行わないが、よりパッケージレベルを細分化して、細かく流通ビジネスを展開することもできる。

【0040】(様々な規格の構成)図9を用いて、代表的なメディアであるCD(図9(a))、DVD(図9(b))、DVB-MHP(図9(c))およびEパッケージ(図9(d))それぞれの規格の構成、概要について説明する。なお、VHSは、物理的な特性と電気信号とから構成され、データ構造を有する他の規格とは大きく異なるので、VHSについての説明は省略する。

【0041】CDは、サンプリング周波数44.1kHzで標準化されたデータと、TOC(Table Of Contents)と呼ばれる各トラック(曲)へのインデックス情報とから構成されている。CD再生機は、TOCを読み込み、ユーザからのリクエスト、例えば「トラック3の再生」など、を受けて、対応する曲のデータを読み出し、DA変換を行って再生する。

【0042】図示はしていないが、CDを改良したVideo CDには、AVストリームと、そのインデック

スとなるPSD(Programable Sequence Descriptor)とが記録されている。Video CD再生機は、ユーザの操作に従って、対応するAVストリームの読み出しとデコードを行って再生する。CDとVideo CDのどちらも、データ構造が規格とされ、再生機器は規格に従い、データ構造を解釈し実行している。

【0043】DVDには仮想マシンという概念が導入された。これはCPUと同じように、演算処理機能とレジスタ(専用メモリ)を有する構成である。データ構造として記述されているシナリオデータに対して、ユーザの操作やレジスタの値によって、プレーヤの動作が異なってくる。

【0044】簡単な例を説明すれば、映画の視聴者が「大人」であるか、「18歳未満の子供」であるかによってストーリーを分けることができる。これは、パレンタルロックと呼ばれる機能である。これによって、映画の中にある性的シーン、暴力シーンを子供に見せないようにカットする。他にもユーザが男性であるか、女性であるかによってストーリーやアングルを変えることもできる。

【0045】DVDでは、静的なデータ構造の他に、仮想マシンとしてプレーヤ(映像再生装置ともいう)の動作モデルを規格として定義している。これによって、プレーヤメーカー間で異なるハードウェアやソフトウェアのプラットフォームや、プレーヤアプリケーションのソフトウェアインプリメンテーションの差異を吸収し、プレーヤ間での互換性を実現する。

【0046】次に、DVB-MHP(Digital Video Broadcasting Multimedia Home Platform)について説明する。DVB-MHPは、欧州で標準化が進んでいる次世代のデジタル放送規格である。この規格の最大の特徴は、Java(登録商標)と呼ばれるミドルウェアを用いていることである。

【0047】Java(登録商標)は、プラットフォーム間での互換性を改善するためにSun Microsystems社が提唱したミドルウェアである。Java(登録商標)を搭載したコンピュータ、機器上では、全てのJava(登録商標)アプリケーションが動作することができ、プラットフォームの垣根を越えてアプリケーションの使用範囲を大きく広げられることが最大の特徴である。

【0048】国内でも、NTT DoCoMoのiモードや、AV機器間のネットワークを目指しているHaviなどで、Java(登録商標)を採用の動きがある。

【0049】DVB-MHPでは、Java(登録商標)の導入とともに、DVB-MHPに特化した、即ち、テレビ放送の映像番組やデータ放送番組を処理するオブジェクトクラスとそのインターフェースを定義して

いる。

【0050】DVB-MHPは、従来の規格と比較すると、静的なデータ構造を決めているのではなく、ミドルウェア上のインターフェースを規格として定義している点で大きく異なっている。

【0051】このため、アプリケーションには、コンピュータプログラムとして作ることが出来る全てのものが使用できる。その反面、アプリケーションを作るシステムが何も設けられていない。そのため、使用されるアプリケーションは、従来のコンテンツビジネスと比較すると、音楽や映画などのAV系よりもコンピュータゲームの方に近い。

【0052】本発明のEパッケージでは、DVB-MHPと同じようにミドルウェアを載せ、プレーヤ上で様々なアプリケーションが動作できるようにしている。しかし、映画という最大のコンテンツを効率良く作るには、従来型の静的なデータ構造と、仮想マシンのようなプレーヤ動作モデルとを有するほうが便利である。

【0053】そこで、本発明のEパッケージは、映画コンテンツ用に静的なデータ構造とプレーヤ動作モデルを定義する。また、このEパッケージは、映画コンテンツの価値を高めるアプリケーションに対してのインターフェースを提供する。

【0054】(プレーヤモデル) 図10はミドルウェア上でのプレーヤモデル概念図である。ここで使用するミドルウェアはJava(登録商標)のようなオブジェクト思考プログラミング言語である。オブジェクト思考プログラミング言語およびその基本クラスそのものについては、その多くが書籍またはインターネット上のホームページにおいて公開されているため、その詳細、特にクラスライブラリ内部での処理は省略する。

【0055】Eパッケージのミドルウェアでは、タイトルや言語設定など様々な機能がクラスおよびそのメンバ関数として定義される。各クラスは実行時にそのインスタンスが生成され、プレーヤアプリケーション等のアプリケーションからアクセスされる。

【0056】ここで使用するクラスを簡単に説明する。図10中の楕円は各クラスのインスタンスを示している。

【0057】「Titleクラス」は、各映画タイトルに相当するEパッケージ専用のクラスである。このクラスは、チャプタなどのシナリオ情報、AVデータのアドレス情報、アプリケーションに対して提供するインターフェース情報等を有する。

【0058】これらの情報は全て再生制御情報ファイル(図中の下段)に記述されている。再生制御情報によって記述された属性は、そのままオブジェクトの属性となる。例えば、Titleインスタンスのレベル属性は、再生制御情報Titleの属性levelによって指定される。詳細は後述する。

【0059】また、Titleクラスは再生制御用のメンバ関数(Methods)を有する。例えば、タイトルの再生は「Play()」関数をコールすることで行われ、再生停止は、「Stop()」関数をコールすることで行われる。

【0060】これらのメンバ関数(Methods)も、再生制御情報によって、その機能が制御される。例えば、TitleインスタンスのSetRate(特殊再生関数)は、再生制御情報の<SETRATE level="">によって、その関数の使用が制限される。詳細は後述する。

【0061】「Audioクラス」は、オーディオストリームに相当するクラスである。このクラスは各オーディオストリーム毎にインスタンス化される。インスタンスは、ストリームの属性や言語情報を持つ。例えば、オーディオストリームの言語情報は、<AUDIO language="Japanese">のようにして再生制御情報で定義される。この属性値は、Audioインスタンスからメンバ関数getLang()によって取り出す事が可能である。

【0062】Eパッケージは、DVDと同様にマルチ言語に対応しており、ユーザは好みの音声ストリームの選択ができる。プレーヤアプリケーションはユーザからのリクエストを受け、対応するインスタンスをTitleクラスインスタンスへ指定(set)する。この時、前述したように各Audioインスタンスのメンバ関数getLang()を使って、対応言語を調べ、ユーザのリクエストにあったAudioインスタンスを選び、Titleインスタンスへ指定(set)する。

【0063】「Subtitleクラス」は、字幕ストリームに相当するクラスであり、Audioクラスとほぼ同様の機能を有している。

【0064】「Socketクラス」は、ネットワークを経由して他のプレーヤ(映像再生装置)およびサーバ等と通信するクラスである。

【0065】「Loaderクラス」は、他のアプリケーションなどを動的に組み込むクラスである。Loaderクラスによって動的に組み込まれるアプリケーションは、再生制御情報ファイルによって定義される。プレーヤアプリケーションを使って他のアプリケーションを再生する場合には、通常、Loaderクラスを使用する。しかし、プレーヤ機能も備えたアプリケーションが実行される場合には、必ずしもLoaderクラスを呼ぶ必要はない。

【0066】「Eventクラス」は、シナリオ内に記述されたイベントトリガを発生するクラスである。例えば、映画の途中でユーザにダイアログメッセージを表示するなどに用いることができる。

【0067】「Cursorクラス」はユーザのカーソル移動をアプリケーションに伝えるクラスである。リモ

コンによるカーソルの移動などをキャッチする。

【0068】「Buttonクラス」、「Canvasクラス」、「Frameクラス」はそれぞれ画面上にボタン、キャンバス、フレームを表示するクラスである。これらのクラスはインスタンスを生成して、画面上に組み込む (add) ことで描画が行われる。

【0069】特にCanvasクラスは、動画を描画するクラスである。このCanvasクラスのインスタンス上にTitleインスタンスを組み込む (add) ことで画面上に動画表示を行うことができる。また、Titleインスタンスを外す (delete) ことで動画表示を終了させることができる。

【0070】「Textクラス」は、画面上にテキスト表示を行うクラスである。コンストラクタによって、Textインスタンスを生成し、Canvasインスタンス上にTextインスタンスをおく (add) ことで画面上に任意のテキストを描画することができる。

【0071】(アプリケーションの例) 上述したプレーヤモデルの構成によって、以下に記すアプリケーションを実現できる。

【0072】図11に示すのはシンプルなDVDプレーヤの例である。図11に示す通り、DVDプレーヤアプリケーションも一つのアプリケーションとしてミドルウェアの上に実装されている。プレーヤアプリケーションは、ミドルウェア上で提供されるクラスライブラリからインスタンスを生成し、そのメンバ関数を呼び出すことで、タイトルの再生を行う。

【0073】例えば、Titleクラスから作られるメニューインスタンスをCanvasインスタンスに設定 (add) することで画面上にメニューを表示し、ユーザからのリクエストを受け付ける。ユーザはカーソルを使い再生するタイトルを選択する。

【0074】ユーザからのリクエストは、Cursorクラスのインスタンスを通してタイトルまたはメニューに届けられる。例えば、メニューでは、ユーザが選択したタイトルに対応するTitleクラスのインスタンスを取得し、Canvasインスタンスに載せ (add)、再生 (play) を実行する。

【0075】図12に示すのはゲームアプリケーションの例である。図12では、プレーヤアプリケーションに代わってゲームアプリケーションが起動されている。ゲームアプリケーションは、パッケージ内のタイトルから任意の画面を選びゲームの背景画面として表示している。ゲームアプリケーションは3Dポリゴン画像を背景画像の上に載せ、ゲームを進行していく。基本的な動作は前述したプレーヤアプリケーションの例と同じであり、アプリケーションプログラムが専用プレーヤアプリケーションでなく、ゲームアプリケーションに代わっている。

【0076】勿論、背景画面を細かく制御して、ゲーム

と同期させた表示を行うことも可能である。

【0077】図13に示すのはタイトル間のリンク構造である。前述したように、ホームサーバ上には多数の映画コンテンツが記録されている。どの映画タイトルが実際に記録されているのかは、各家庭によってバラバラであり、図13に示すようにタイトル間に跨る構造は一意に規定することはできない。

【0078】そこで、本発明のEパッケージでは、各タイトル毎にリンク先の情報を有し、実際に有効なリンク先だけが再生時に有効となる構成を取っている。

【0079】例えば、「Title1」は、「Title2」、「Title3」、「Title5」、「Title6」のリンク情報を有している。一方、ホームサーバ上には「Title5」が存在しない。この場合、「Title1」再生時の有効リンクは、「Title2」、「Title3」または「Title6」になる。このように動的に再生可能なリンクだけを選び出すことを可能としている。

【0080】(規格の構成) 図14は規格の構成を示す図である。図14に示すようにEパッケージ規格は、プレーヤモデル、データ構造、AVデータの主に3つのパートから構成されている。

【0081】プレーヤモデルは、オブジェクト指向プログラミング言語のクラスライブラリとして構成されており、再生制御情報に基づいてメニューやタイトルなどの機能をそれぞれインスタンス化してアプリケーションに提供する。

【0082】データ構造は、図14に示すように、パッケージ全体を管理するパッケージ情報、メニューを記述したメニュー情報、各タイトル毎にシナリオを記述したタイトル情報、ストリーム毎に属性やアクセスするためのアドレスなどを記述したストリーム情報等から構成されている。以下詳細を説明する。

【0083】まず、図15を用いて、パッケージのディレクトリ、ファイル構造を説明する。

【0084】Eパッケージは、DVDと同じように単体の光ディスクとして流通する場合と、ネットワークを通じて電子的に配信され、HDDに蓄積される場合とがある。ここで説明するディレクトリ (フォルダという場合もある)、ファイル構造はどちらも共通に使われる形式である。

【0085】Eパッケージは、DVDと同様にファイルシステムを導入している。Eパッケージのファイルシステムでは、「PACKAGE」ディレクトリがルートディレクトリの直下に置かれている。このディレクトリはEパッケージ専用のディレクトリであって、他のアプリケーション、例えば従来のDVDデータなどは置かれていない。「PACKAGE」ディレクトリの下に更にサブディレクトリが置かれ、それぞれが一つのパッケージに対応している。図15での「abc」、「あいう」が

それに対応する。

【0086】サブディレクトリの中に各管理情報に対応するファイルと、ストリームデータが置かれている。最初の「package.xml」は予約ファイルであって、前述の「パッケージ情報」が記録されている。その他には、メニューを記述した「menu.xml」、タイトルを記述した「title1.xml」、「title2.xml」、ストリーム情報を記録した「stream1.xml」、「stream2.xml」が置かれている。

【0087】(データ構造詳細)図16はパッケージ情報「package.xml」の詳細を示している。データの記述は前述した通り、XMLの記述方式に従い、タグ<PACKAGE>で囲まれた中にパッケージ情報が記述され、以下の情報を有している。

<GENERAL> 一般情報
バージョン情報 (version)
<ACCESS> アクセス制限情報
地域情報 (region)

【0088】Eパッケージは、この地域情報を用いて映像コンテンツへのアクセス制限を行うことにより、映像コンテンツを再生できる地域を限定できる。この地域情報に再生可能な地域を順次増やしていくか、又は地域ごとに該当する地域情報を付与することによって、映画タイトルのタイムシフト、例えば北米を皮切りに日本および欧州、アジア、中国の順にタイトルを供給していくことが可能である。地域情報 (region) は、それぞれ、「US」、「Japan」、「EU」、「Asia」、「China」などの値を持つ。

【0089】<UPDATE> 更新予告情報

日時情報 (data)

自動取得の実施フラグ (auto)

この更新予告情報には、シナリオ他、映画タイトルの自動更新予定が記述される。プレーヤ (映像再生装置) は、この情報に基づいて自動的にインターネットを通じて新しい情報への更新ができる。

【0090】<INTERNET> インターネットホームページ情報

URL (URL)

このインターネットホームページ情報は、関連する情報が記載されているインターネットのホームページアドレス情報である。ユーザからインターネットアクセスの要求があった場合に、この情報が示すアドレスへアクセスする。また、前述の更新予告でも上記のアドレスを基に情報の取得を行う。

【0091】<MENU> メニュー情報

メニュー情報ファイル (menu)

メニュー情報は、メニュー情報ファイルを指定する。指定されたファイルの中にメニュー情報が記述される。

【0092】<TITLE_LIST> タイトルリス

ト

パッケージで取り扱うタイトルは<TITLE_LIST>タグに囲まれた中で<TITLE>タグによって記述される。

<TITLE> タイトル情報

タイトル番号情報 (number)

タイトル情報ファイル (file)

タイトル情報には、各タイトルへのリンク情報が記述される。個々のタイトル自身は、指定されたタイトル情報ファイル内に記述されている。

【0093】図17はメニュー情報「menu.xml」の詳細を記している。タグ<MENU>で囲まれた中に以下に示すメニュー情報が記述されている。

<MENU_PAGE> メニューページ情報

ページ番号 (page)

背景画像情報 (image)

メニューページ情報は、メニュー画面が複数あるマルチページメニューに関する情報である。例えば表示したいタイトルが100以上と大量にあるために1枚のページだけでは表示しきれない場合には、マルチページメニューが使われる。

【0094】<TITLE> タイトル情報

横座標 (column)

縦座標 (row)

タイトル番号 (title)

オブジェクト名 (object)

タイトル名 (<TITLE>タグに囲まれて記述される)

タイトル情報タグ毎に、各タイトルの情報が記される。プレーヤアプリケーションはこの情報を基にメニュー表示を行う。また、オブジェクトで指定される部品がグラフィカル・ユーザ・インターフェースとして、画面上に表示される。この部品はミドルウェア上のクラスライブラリの機能として提供されている。

【0095】例えば、図17のように、オブジェクト (object) がボタン (button) だった場合は、ミドルウェアのグラフィックライブラリが提供するボタンオブジェクトがメニュー上に表示される。表示位置は、横座標 (column) および縦座標 (row) によって示され、また、タイトル名がボタン上に表示される。

【0096】図18はタイトル情報「title1.xml」他、の詳細を記している。タグ<TITLE>で囲まれた中に以下に示すタイトル情報が記述されている。

<TITLE> タイトル情報

タイトル番号 (title)

レベル (level)

レベルは、本タイトルの再生レベルを示している。前述したように、Eパッケージではパッケージの再生レベル

を設けることで、ユーザの購入状況に応じて再生可能レベルを分けることができる。具体的には、レベル (level) は、フルパッケージ (full)、制限つきパッケージ (restricted)、無料パッケージ (free) のいずれかの値を持つ。一方、プレーヤの持つステータス (Status) が、「フル再生可能 (full playback)」である場合は、全てのパッケージの再生が可能であり、ステータス (Status) が、「制限付き再生可 (restricted)」である場合 (restricted playback) は、制限つきパッケージ (restricted) または無料パッケージ (free) の何れかが再生できる。プレーヤ側の属性が、「無料パッケージのみ (free only)」の場合は、無料パッケージ (free) のみ再生可能である。

【0097】ここでは、パッケージの種類を3種類に限定したが、分類数は本質的な問題では無く、2種類、4種類またはそれ以上の種類に分類し、再生制限を行うことはもちろん可能である。また、分類の仕方や名称も上述の限りに限定されるものではない。

【0098】<LINK_LIST> リンクリスト
本タイトル中で発生するリンク一覧を定義する。

【0099】<LINK>情報
識別情報 (ID)

リンク先パッケージ情報 (package)

リンク先タイトル情報 (title)

リンク先チャプタ情報 (chapter)

リンク先時刻情報 (time)

リンク情報がリンク情報タグ毎に記される。各リンク情報は後述するタイムライン情報内で実際に使用される。リンク情報は、タイトル起動時にリンク先の有効、無効をプレーヤが自動検出するために定義されている。

【0100】<CHAPTER_LIST> チャプタリスト

<CHAPTER> チャプタ情報

開始時刻情報 (in)

終了時刻情報 (out)

再生ストリーム情報 (video)

再生字幕情報 (subtitle)

チャプタはタイトル情報内でエントリされる。

【0101】<TIMELINE> タイムライン情報
時間軸上に展開されるイベントなどの情報は、タイムライン情報タグ内に記述される。記述される情報は以下の通りである。

<BRANCH> 分岐情報

レベル情報 (level)

メッセージ情報 (message)

識別情報 (ID)

有効区間開始時刻情報 (in)

有効区間終了時刻情報 (out)

分岐先タイトル (jump)

【0102】レベル情報 (level) は、前述した通り映像再生装置のステータス (Status) に応じて処理を行うか否かを示すフラグである。例えば、映像再生装置のステータス (Status) が「無料パッケージのみ (free only)」で、レベル情報 (level) が「フルパッケージ (full)」であった場合などには、当該分岐タグ (BRANCH) は無視される。また、識別情報 (ID) は前述したLINK情報の識別情報 (ID) に対応している。

【0103】プレーヤモデルはユーザからの分岐要求を受けた場合に、対応するLINK情報に記述されている場所の再生を開始する。

【0104】<MESSAGE> メッセージ情報

レベル情報 (level)

メッセージ情報 (message)

識別情報 (ID)

有効区間開始時刻情報 (in)

有効区間終了時刻情報 (out)

メッセージタグによって記述されるメッセージは、プレーヤのOSDによって字幕表示される。

【0105】<TRIGGER> イベントトリガ情報

レベル情報 (level)

イベント情報 (event)

識別情報 (ID)

発生時刻情報 (time)

発生時刻になった時点で、アプリケーションに対してイベントを投げる。内容はイベント情報 (event) に記され、そのままアプリケーションに渡される。

【0106】<INTERFACE> インターフェース情報

<PLAY> 再生機能制御タグ

<STOP> 停止機能制御タグ

<SETRATE> 特再機能制御タグ

<SETTIME> 飛び込み再生機能制御タグ

<SETAUDIO> 音声設定機能制御タグ

<SETSUBTITLE> 字幕設定機能制御タグ

【0107】インターフェース情報 (<INTERFACE>) は、上述する幾つかのプレーヤ機能制御タグを持つ。個々のタグはTitleインスタンスのメンバ関数、play、stop、setRate、setTime、setAudio、setSubtitleにそれぞれ対応する。また、個々のタグは属性レベル (level) を持ち、パッケージのレベル (level) と同様に "full"、"restricted" または "free" のいずれかの値を持つ。

【0108】例えば、レベル (level) が "full" の場合は、対応するTitleインスタンスのメンバ関数の使用が制限される。この場合、映像再生装置が持つステータス (Status) が「フル再生可能 (f

ull playback)」の場合にのみ当該関数の使用が可能である。各関数のレベル (level) と、プレーヤアプリケーションのステータス (Status) の関係は前述したパッケージでのレベル (level) と同じである。

【0109】図19はストリーム情報「stream1.xml」他、の詳細を記している。タグ<STREAM>で囲まれた中に以下に示すタイトル情報が記述されている。

<STREAM> ストリーム情報

ファイル情報 (file)

ファイル情報は再生対象のストリームファイル名を記述する。

【0110】<ATTRIBUTE> 属性情報

以下に記すビデオおよびオーディオの属性情報は属性情報タグに囲まれて記述される。

【0111】<VIDEO> ビデオ属性情報

圧縮情報 (coding)

解像度情報 (resolution)

アスペクト比情報 (aspect)

【0112】<AUDIO> オーディオ属性情報

圧縮情報 (coding)

ビットレート情報 (bitrate)

チャンネル数情報 (channel)

言語情報 (language)

【0113】<TIMEMAP> タイムマップ情報

タイムマップ情報には、VOBU (詳細は後述する) 毎の時間およびサイズ情報が記述されている。各VOBUとしてエントリされているユニットの再生時間長 (フレーム数) とデータサイズ (バイト数) が記されている。

【0114】再生ストリームの任意の時刻へ飛び込み、再生を行う場合、タイムマップ情報の各エントリの時間情報を加算して対象となるVOBUを検出し、同様にVOBUのサイズ情報を加算した結果に基づいて、ファイル内でのシーク先を導くことが可能になる。このように、タイムマップ情報は、ストリーム内の時間情報とアドレス情報の変換を行うフィルタの役割を果たしている。

【0115】<ENTRY> エントリ情報

時間情報 (duration)

サイズ情報 (size)

【0116】図20は、字幕情報「subtitle1.xml」他、の詳細を記した図である。

【0117】タグ<SUBTITLE>に囲まれた下記の情報によって各言語毎の字幕が記述されている。

<LANGUAGE> 言語情報

言語情報 (language)

キャラクタ情報 (character)

フォント情報 (font)

カラー (color)

斜体 (italic)

太字 (bold)

下線 (underline)

言語情報タグの属性値として、英語、日本語などを示す言語情報、シフトJISなどのキャラクタ情報、明朝体などのフォント情報と、表示文字の修飾情報が記述される。

【0118】<TEXT> テキスト情報

表示開始時刻情報 (in)

表示終了時刻情報 (out)

テキスト

【0119】(ストリーム構造) 図21を用いてストリームの詳細を説明する。本実施例で使用するストリームは、MPEG-2と呼ばれる国際規格ISO/IEC13818に基づいている。MPEG-2は、ビデオストリーム、オーディオストリームと、それらを多重化 (一つのストリームに束ねる) するシステムストリームとから構成されている。

【0120】ビデオデータは、Iピクチャ (フレーム内符号化)、Pピクチャ (時間予測符号化) およびBピクチャ (双方向時間予測符号化) を有するGOP構造に圧縮される。それぞれのピクチャ間での参照関係は図21の通りである。

【0121】圧縮されたビデオデータは、パケット化され、次いでバック化されてオーディオデータと多重化され、1本のシステムストリームを形成する。

【0122】多重化層では、GOPを基準 (GOP先頭を含むバックから次のGOP先頭を含むバックの直前まで) としてVOBUを形成する。GOPはビデオ層で定義されている言葉であり、システム層での定義に当てはめることができないため、VOBUを導入している。

【0123】また、本実施例の中では、MPEG-2システムストリームのことをVOB (Video Object) という。

【0124】(プレーヤ構造) 図22は、映像再生装置のブロック構成図である。映像再生装置は、STBなどの外部チューナからデータを受け取る受信手段 (101)、データを記録する蓄積媒体 (102)、CPU (103)、プログラムメモリ (104)、作業用ワークメモリ (105)、ストリームをデコードするデコーダ (106)、モニタおよびスピーカへ表示出力する表示手段 (107)、ユーザからの要求を受けるI/F (108) から構成されている。CPU (103) はその内部にクロックを持ち、日時情報を有しており、また、作業用ワークメモリ (105) には、映像再生装置の再生制御ステータス情報 (full/restricted/free) が格納されている。

【0125】(クラスライブラリ詳細) 図23は、Eパッケージの映像再生装置のソフトウェア構成図である。オペレーティングシステム (203、以後OSと略す

る)を中心に、OSの下には、ファイルシステムドライバ(201)と、デバイスドライバ(202)とを有する。ファイルシステムドライバ(201)は、ディスク上のデータをファイルまたはディレクトリ構造を用いてアプリケーションへのアクセス環境を提供する。デバイスドライバ(202)は、コンピュータのハードウェアデバイスであるデコーダやグラフィックカードを制御する。

【0126】また、OS上には、ミドルウェア(204)が乗る。例えばJava(登録商標)の場合は、Java(登録商標)の仮想マシン(Virtual Machine、以下Java(登録商標)VMと称する)とクラスライブラリが置かれる。このクラスライブラリとして、Eパッケージ用のクラスライブラリ(205)も置かれる。

【0127】標準のクラスライブラリおよびEパッケージ用のクラスライブラリは、アプリケーションに対してクラスやそのメンバ関数などをプログラミング環境として提供する。

【0128】アプリケーションとして、Eパッケージ専用のプレーヤアプリケーション(206)の他にサードパーティから提供される外部アプリケーション(207)がそれぞれ動作する。

【0129】図24は、ミドルウェア内でのEパッケージクラス構成図である。ミドルウェア内にはEパッケージ用として、「Packageクラス」「Titleクラス」「Menuクラス」「Audioクラス」「Subtitleクラス」「Eventクラス」「Linkクラス」「Cursorクラス」「Statusクラス」などが用意されている。以下、個々に説明する。

【0130】「Packageクラス」Packageクラスは、最初に呼び出されるクラスである。パッケージ情報package.xmlを基にインスタンスが生成される。

【0131】図25はPackageクラスの処理を示している。コンストラクタPackage(package)は、package.xmlを読み込み、Packageインスタンスの属性値を取得する(2501)。前述したように、インスタンスの属性値は、全て対象の管理情報ファイル内に記述されている。

【0132】次に、リージョン情報(region)、レベル情報(level)日時情報(expire)からパッケージの再生が可能か否かの認証処理を行う(2502)。再生不可(prohibited)の場合は、アプリケーションに対してエラーが返され、終了する(2503)。

【0133】認証プロセスを通過(permitte d)した場合は、アップデートチェックを行う(2504)。

【0134】アップデートタグ(<UPDATE>)の

日時情報(date)をCPUの日時情報と比較して、アップデート予告日を経過していた場合、且つ、自動アップデート情報(auto)が"yes"の場合、インターネットからのダウンロードを実行(2505)して、新しい再生制御情報で再生を再開する(2501)。

【0135】アップデートチェック(2504)でダウンロードを実行しない場合、Menuインスタンス生成(2506)、Titleインスタンス生成(2507)をそれぞれ実行する。

【0136】Packageインスタンスは、getMenuメンバ関数(図25(b))、getTitleメンバ関数(図25(c))を有する。Packageインスタンス生成後、アプリケーションはこの関数をコールして、Menuおよび各Titleインスタンスをそれぞれ入手することができる。

【0137】「Titleクラス」Titleクラスは、タイトルの再生制御を行うクラスである。各タイトル毎にインスタンスが生成され、インスタンスをCanvasインスタンス上に配置(add)することで画面への描画が行われる。また、メンバ関数のコールによってタイトル再生が制御される。

【0138】図26および図33はTitleクラスの処理を示している。コンストラクタTitle(title)は、起動されると同時にtitle.xmlを読み込み(2601)、Link_LISTに基づきLinkリストを内部に生成する(2602)。この時、対象タイトルがアクセス可能な場所にあるかをチェックし、アクセス不可能の場合はリストから削除する。具体的には、ネットワークプロトコルなどを使って当該ファイルが存在するのを確認するが、本発明とは直接関係しないので、詳細な説明は省略する。

【0139】次に、Chapterリストを生成(2603)し、Chapterが参照するストリームの属性情報ファイル(例えば、stream.xml)を読み込み(2604)、Audio、Subtitleインスタンスを生成する(2605)。

【0140】次に、TIMELINE情報に基づき、Timelineリストを生成(2606)し、INTERFACE情報に基づき、関数リストを生成(2607)し、最後にCursorインスタンスを生成(2608)し、リモコン(I/F)からの要求に対応できるように準備する。

【0141】Titleクラスは、色々なメンバ関数を持っている。再生(play)、停止(stop)、再生速度設定(setRate)、再生位置設定(setTime)は、AV再生を直接コントロールする関数である。これらはデコーダが提供する機能をそのままアプリケーションに提供する。例えば、playは、アプリケーションからコールされると、関数の有効/無効をチェックし、有効の場合は、デコーダに再生開始を指示す

る。

【0142】例えば、アプリケーションから再生関数 (play) が呼ばれた場合を考える。この時、再生関数 (play) は、プレーヤの再生可能状態 (full playback/restricted playback/free only) と関数リストから当該

	level = full	restricted	free
Status = full playback	有効	有効	有効
restricted playback	無効	有効	有効
free only	無効	無効	有効

上記の表は、Titleインスタンスの関数の有効無効だけでなく、パッケージ自身のレベル (level) に対して再生が可能かを判断する基準にも一致する。

【0144】音声、字幕の制御は、タイトル内でのストリーム、即ち、言語毎の属性値を持つインスタンスを取得する getAudio、getSubtitle と、再生するストリームを設定する setAudio、setSubtitle とがある。

【0145】getAudio および getSubtitle は、Title コンストラクタで生成した Audio インスタンス、Subtitle インスタンスをそれぞれの返り値としてアプリケーションに渡す (2621)。アプリケーションは、ここで得られたインスタンスをそのまま setAudio または setSubtitle の引数として使い再生ストリームを設定する。

【0146】setAudio および setSubtitle 関数では、最初に当該関数が有効かをチェックする (2631)。具体的には、映像再生装置の再生可能状態 (full playback/restricted playback/free only) と関数リストの当該関数の使用制限を比較する。当該関数が使用可能である場合は、受け取ったインスタンスの属性値に従い、デコーダに再生ストリーム属性値を設定する (2632)。一方、当該関数が使用不可能である場合は、関数の処理を中止する。

【0147】映像再生装置のステータスと関数との比較は前述した表と同じである。

【0148】Title クラスは、その他にもイベント処理起動用に enableEvent を、タイトル間リンク処理起動用に enableLink 関数を持つ。

【0149】enableEvent 関数は、Title 情報で記されたタイムライン情報 (<TIMELINE>)、即ち、分岐情報 (<BRANCH>)、メッセージ情報 (<Message>) 及びイベントトリガ情報 (<TRIGGER>) を処理する。enableEvent 関数は、コールされると内部でスレッドを起動する (3301)。起動されたスレッドは、以下のループ処理を行い続ける。再生時刻情報を監視し、タイムラインリストにある各イベント、例えば分岐情報 (<BRANCH>)、メッセージ情報 (<Message>

関数の使用制限を比較する (2611)。当該関数が使用可能である場合は、関数の実行を開始する (2612)。一方、当該関数が使用不可能である場合は、関数の処理を中止する。

【0143】ここで、関数の有効/無効の関係を下記の表に整理する。

>)、または、イベントトリガ情報 (<TRIGGER>) が指定する有効時刻になっているかをチェックする (3302)。有効時刻になっている場合は、映像再生装置の再生可能情報 (Status) と比較 (3303) して各イベントが当該映像再生装置で実行可能か否かを判断する。

【0150】各イベントが実行可能な場合は、イベントの種類がユーザからのリクエストを必要とする分岐 (BRANCH) か否かを確認する (3304)。

【0151】分岐 (BRANCH) の場合は、ユーザからのリクエスト受け付け (3305) へと進み、分岐 (BRANCH) の有効時間 (out) によって指定) までリクエストを待ちつづけるようにループする (3306)。リクエストを受けずに有効時間 (out) が切れた場合は、再びループの先頭 (3302) まで戻る。有効時間 (out) までにユーザからのリクエストを受け付けた場合は、分岐先タイトル Title インスタンス (jump で指定) を生成し、当該タイトルの再生へと進む (3307)。

【0152】ステップ 3304 で、当該処理が分岐 (BRANCH) で無い場合、即ち、メッセージ (MESSAGE) またはイベントトリガ (TRIGGER) と判断された場合は、ステップ 3308 へ進み、当該処理がメッセージ (MESSAGE) かイベントトリガ (TRIGGER) かを判断する。該処理がメッセージ (MESSAGE) の場合は、指定されたメッセージ情報 (message) から Text インスタンスを生成し (3309)、Text インスタンスを Canvas インスタンス上に表示 (add) する (3310)。また、メッセージ表示期間 (out) まで待ち (3311)、表示期間終了 (out) と同時に Text インスタンスを消去 (Canvas インスタンスから delete) し (3312)、ループ先頭 (3302) まで戻る。

【0153】ステップ 3308 で、当該処理が TRIGGER と判断された場合は、Event インスタンスを生成して (3313)、アプリケーションによってインプリメントされている関数 (eventExec) を実行し (3314)、ループ先頭 (3302) まで戻る。

【0154】「Menu クラス」Menu クラスは、Title クラスから派生したメニューを表示するクラス

である。タイトルと同じMenu毎にインスタンス化され、Canvasクラスに置くことで、画面表示がされる。

【0155】図27は、Menuクラスの処理を示している。コンストラクタMenu()は、メニュー情報ファイルmenu.xmlを読み込み(2701)、Titleクラスとしての処理(2702)を経て、メニューページを生成(2703)、先頭メニューページを表示(2704)し、Cursorからのイベントを処理するmenuThreadを起動する(2705)(図27(a))。

【0156】各ページとして表示する内容は、図17で説明したようにMENU情報内のメニューページ(<MENU_PAGE>)によって記述されている。メニューページ内のタイトル(<TITLE>)に基づいてボタンが作られ、画面上に表示される。

【0157】マルチページメニューの場合は、メンバ関数nextPageによって次ページへ(2711)(図27(b))、prevPageによって前ページ(2721)の表示へ移動する(図27(c))。また、タイトルが選択された場合、selectedTitleによってアプリケーションに対して選択されたタイトル情報を通知する(2731)(図27(d))。

【0158】menuThread関数では、スレッドを起動(2741)し、Cursorインスタンスからのイベントを受け付ける(2742)。Cursorインスタンスからのイベントを受け付けると、イベントがタイトル選択であるかをチェックし(2743)、タイトル選択であった場合は、selectedTitle関数を呼び出し(2744)、アプリケーションに対して、選択タイトルを通知する。

【0159】ステップ2743で、タイトル選択でなかった場合は、次にページ移動があったかをチェックし(2745)、真であったばあいには、ページ移動が次ページへか、前ページへかを判断し(2746)、それぞれnextPageの起動(2747)、prevPageの起動(2748)を行う(図27(e))。

【0160】「Audioクラス」および「Subtitleクラス」

Audioクラスは、各オーディオストリーム毎に属性値を有するクラスである。タイトル内で、例えば使用可能なオーディオストリームが2本あれば、Audioクラスのインスタンスが2つ生成される。どちらから一方のインスタンスをTitleクラスのsetAudio関数に設定することで、再生するオーディオストリームの設定が行われる。

【0161】図28は、Audioクラスの処理を示している。コンストラクタAudio()は、ストリーム属性情報ファイルstream.xmlを読み込み(2801)、インスタンス内に属性値を確保する(280

2)(図28(a))。

【0162】また、Audioクラスは、メンバ関数getLang(図28(b))でインスタンスの持つ言語情報、即ちストリームの対応する言語と、getCoding(図28(c))でインスタンスの持つ圧縮情報、即ちストリームの圧縮方式(2811)と、getChsでインスタンスの持つチャンネル情報、即ちストリームのチャンネル数(2821)とをそれぞれアプリケーションに返す。

【0163】SubtitleクラスもTitleクラスと同様の機能を有する。

【0164】「Eventクラス」および「Linkクラス」

Eventクラスは、タイトル内でのイベント生成を行うクラスであり、Linkクラスは、タイトル内でのタイトル間リンク情報用イベントの生成を行うクラスである。

【0165】図29は、EventクラスおよびLinkクラスの処理を示している。Eventクラスのコンストラクタは、引数を基にEvent属性を設定する(2901)(図29(a))。

【0166】Eventクラスのメンバ関数execEventは、アプリケーションによってオーバーライトされる関数である(図29(b))。即ち、execEventの起動はイベントハンドラを起動する(2911)。execEventは引数にID(id)を持ち、このIDによってアプリケーションはどのイベントトリガ(TRIGGER)がかけられたのかを識別し、以後の処理を分岐して行うことが可能である。

【0167】Linkクラスのコンストラクタは、引数で与えられたTitleのインスタンスを生成する(図29(c))。

【0168】Linkクラスのメンバ関数であるnotifyLinkは、execEventと同じくアプリケーションにオーバーライトされる関数であり、この関数を用いてアプリケーションへTitleインスタンスを渡し、イベント処理を実行する(図29(d))。

【0169】「Cursorクラス」Cursorクラスは、画面上でのカーソルを処理するクラスである。図34を使って説明する。

【0170】Cursorクラスのコンストラクタは、最初に位置情報を生成/初期化し(3401)、リモコンとの通信を開始し(3402)、カーソル処理のスレッドCursorThreadを起動する(3403)(図34(a))。

【0171】カーソル処理スレッドCursorThreadでは、最初にスレッドを起こし(3411)、処理ループに入る。処理ループでは、カーソルの移動が起きたかをチェックし(3412)、移動が起きた場合には、moved関数を呼び(3413)、位置情報を更

新する。移動がなかった場合、もしくはステップ3413の後に、選択実行が行われたか、言い換えれば、カーソルが選択しているボタンに対して、ユーザが選択実行を行ったかを検出し(3414)、選択実行が行われた場合は、selected関数を呼び(3415)、現在のTitleへ通知を行う(図34(b))。

【0172】moved関数では、指定された引数に基づき、位置情報を修正し(3421)(図34(c))、selected関数では、選択実行要求があったことをTitleインスタンスへ通知する(3431)(図34(d))。

【0173】「Statusクラス」Statusクラスは、映像再生装置のステータスを表現するクラスである。このクラスは映像再生装置、またはシステムに対して一意に存在するものであって、その都度インスタンス化はされない。アプリケーションからはクラスとしてそのままアクセスが行われる。

【0174】アクセスは、アプリケーションがステータスを知るgetStatusや有効期間を知るgetPeriodなどが用意されている。

【0175】図35に示すように、Statusクラスはパラメータで指定される値に基づき、Status情報およびPeriod情報を内部に生成する(3501)(図35(a))。メンバ関数getStatusはアプリケーションにStatus情報を返し(3511)(図35(b))、メンバ関数getPeriodはPeriod情報をアプリケーションに返す(3521)(図35(c))。

【0176】「Frameクラス」および「Canvasクラス」

画面構成をするクラスである。Frameクラスは画面表示の元になるクラスで、例えばWindows(登録商標) OSで見られるウィンドウに相当する。Frameインスタンス内には動画再生を行うCanvasインスタンスが置かれる。

【0177】図36を用いてCanvasクラスを説明する。Canvasクラスは、コンストラクタにおいて、映像データをオーバーレイ表示するFrameインスタンスを生成する(3601)。続いて、デコーダの初期化(3602)、オーバーレイ、即ちグラフィック機能の初期化(3603)を行う(図36(a))。デコーダの初期化処理および、グラフィック機能の初期化処理は、下位のオペレーティングシステム(OS)や、ハードウェアに依存する処理であり、また、本発明とは本質的に関連がないので、詳細な説明は省略する。

【0178】Canvasインスタンスによって、画面上に描画するウィンドウが表示されるが、実際の画像描画は、メンバ関数addによって実行される。関数addはTitleインスタンスを引数として呼ばれ、Titleインスタンスが持つ、ストリーム情報の読み出し

(3611)、デコーダの設定(3612)を行う。続いてデコーダにデコード処理を開始(3613)させ、オーバーレイにデコード画像の描画を開始(3614)させる(図36(b))。

【0179】また、Canvasクラスは、メンバ関数setSizeを持ち、この関数によってCanvasのサイズを変更できる。内部の処理としては、Frameインスタンスのサイズ変更(3621)と、オーバーレイする表示サイズの変更(3622)をそれぞれ行っている(図36(c))。

【0180】(プレーヤ再生処理)次に、プレーヤアプリケーションとしての再生処理について説明する。図30はプレーヤ再生処理のフローである。プレーヤアプリケーションは、起動後(3001)、下記のようにCanvasインスタンスを生成して、ビデオ表示ウィンドウを生成する(3002)。Canvasインスタンス生成の内部動作は図36で説明した通りである。Canvas objCanvas = new Canvas();

【0181】上記の記述は、Java(登録商標)言語に基づいている。左端のCanvasはクラス宣言であり、続くobjCanvasがCanvasクラスのオブジェクト(インスタンス)であることを宣言している。また、new Canvas()は、Canvasクラスのコンストラクタを呼び、これによって、objCanvasが生成される。

【0182】次にユーザからのパッケージ選択を待ち(3003)、パッケージ選択後に、下記のようにPackageインスタンスの生成(3004)、メニューインスタンスの取得(3005)を行い、メニューを表示する(3006)。また、パッケージインスタンスの生成は、図25で説明した通りである。

```
Package objPackage = new
Package(package); Menu obj
Menu = objPackage.getMenu
(); objCanvas.add(objMenu
);
```

【0183】メニューは図31に示すように、背景画像とタイトル情報表示(テキスト)から構成されている。タイトルの選択(3007)は、カーソルをリモコンで移動させ、目的のタイトル上で「選択」を行うことによって実施される。

【0184】カーソルは、リモコンキー(上下左右)によって移動する。カーソルの移動や、選択実行は、図34で説明したように、スレッドとして起動されているCursorThread関数で検出され、処理される。

【0185】例えば、次ページの移動が選択された場合は、Cursorインスタンスのselected関数が呼ばれ、Menuインスタンスはページ移動要求があったことを知る。次にMenuインスタンスはnext

Page関数をコールして、メニューを次ページへと送る。

【0186】また、カーソルがタイトル4上で「選択」された場合、Cursorインスタンスの(selected)を通してタイトルが選択されたことをMenuインスタンスは取得する。次いで、selectedTitle関数がアプリケーションに対してタイトルが選択をされたことを通知し、アプリケーションはタイトル再生のステップ(3008以降)へと移行する。

【0187】プレーヤアプリケーションは、選択されたタイトル情報を引数に、PackageインスタンスのgetTitle関数をコールし、Titleインスタンスを取得する(3008)。次いで、プレーヤアプリケーションは、取得したTitleインスタンスのplay関数をコールしてタイトル再生を開始し(3009)、enableEvent関数をコールしてイベント用スレッドを起動する(3010)。

```
Title objTitle = objPackage.getTitle(title); objTitle.play(); objTitle.enableEvent();
```

【0188】以後、タイトル再生終了まで、イベント発生の確認(3011)、イベント発生時のイベント処理(3012)、タイトル再生終了の確認(3013)を繰り返す。タイトル再生終了が確認されたら、プレーヤアプリケーションでの処理を終了する(3014)。

【0189】また、ステップ3012のイベント処理は、図33で説明した通りである。図32を用いてタイトル再生中のタイトル間ジャンプ処理を説明する。タイトル1の中には図示するようにタイトル2への分岐受付期間が設けられている。分岐受付期間は、タイトル1の情報ファイルtitle1.xml内のTIMELINE情報内のBRANCHタグとその属性値で定義されている。

【0190】この分岐受付期間中、図32下に示すようにメッセージが表示され、ユーザが「選択」キーを押すとリンク先のタイトル2へと遷移する。

【0191】分岐受付期間(分岐情報(<BRANCH>)の属性値inからoutで指定される区間)に入ると(図33、3302)、当該分岐(BRANCH)処理のStatusと映像再生装置の持つStatus(Status.getStatus())によって取得)を比較して(前述した表の通り)、処理可能かを判断し(図33、3303)、BRANCHであることの確認(図33、3304)を経てユーザからのリクエスト待ちループ(図33、3305および3306)に入る。

【0192】ユーザからの選択実行リクエストは、Cursorインスタンスを通して受け取る(図34、3414から3415)。ユーザからの選択実行リクエスト

があった場合は、新たなTitleインスタンスを生成して、次のタイトル(図32ではタイトル2)の再生を開始する(図34、3307)。また、分岐受付期間(out)までにユーザからの選択実行リクエストがなかった場合には、タイムアウトの検出を経て(図33、3306)、分岐(BRANCH)処理を終了する。

【0193】図37は、ゲームアプリケーションの例である。本例では、プレーヤアプリケーションの代わりにゲームアプリケーションを起動する(3701)。プレーヤアプリケーションと同様に、Canvasインスタンスを生成して、ビデオ表示ウィンドウを生成する(3702)。Canvasインスタンス生成の内部動作は図36で説明した通りである。Canvas objCanvas = new Canvas();

【0194】ゲームアプリケーションはゲームを開始し(3703)、ゲームアプリケーションが使用するPackageインスタンスの取得(3704)とTitleインスタンスの取得(3705)を行う。取得したTitleインスタンスのplay関数をコールしてタイトル再生を開始し(3706)、enableEvent関数をコールしてイベント用スレッドを起動する(3707)。

```
Package objPackage = new Package(package); Title objTitle = objPackage.getTitle(title); objTitle.play(); objTitle.enableEvent();
```

【0195】以後、ゲーム終了まで、イベント発生の確認(3708)、イベント発生時のイベント処理(3709)、タイトル再生終了の確認(3710)を繰り返す。ゲーム終了が確認されたら、ゲームアプリケーションの処理を終了する(3711)。

【0196】ゲームアプリケーションの場合、イベントトリガを使ってゲームとAV再生を同期させることができる。例えば、図18で説明したタイトル情報(<TITLE>)内のタイムライン情報(<TIMELINE>)に、以下のようなイベントトリガ(<TRIGGER>)を入れておく。

```
<TRIGGER level="full" id="1" event="1" time="00:01:00:00"/>
```

【0197】時刻00:01:00:00(1分)になると、イベントスレッドは対象時刻(time)になったことを認識し(図33、3302)、ステータス(Status)確認(図33、3303)、分岐(Branch)確認(図33、3304)、メッセージ(Message)確認(図33、3308)を経て、Event生成(図33、3313)、execEvent起動(図33、3314)が実行される。

【0198】起動されるメンバ関数 `execEvent` は、ゲームアプリケーションによってオーバーライトされており、ゲームアプリケーションは、`execEvent` から得た `id` に基づいて、ゲーム側での同期処理を行うことが可能である。

【0199】図38および図39は、パッケージと映像再生装置との間でのステータス (Status) または有効期限 (Expire) による再生制限を解除するためにネットワークを介してサーバから Status または Expire 情報を更新する例を説明した図である。

【0200】図25で説明した通り、パッケージの持つ再生レベル (level) および有効期限 (expire) と、映像再生装置が持つステータス (Status) および日時情報をそれぞれ比較して、パッケージを再生できるかを認証 (図25、2502) する。再生可能であれば2504以降のステップを処理し、再生不可能であればステップ2503によって再生処理を終了させている。

【0201】図25の再生処理終了 (2503) に代わって、映像再生装置のステータス (Status) またはパッケージの有効期限 (expire) を更新する処理を行ってもよい。

【0202】図39は、映像再生装置のステータス (Status) がパッケージのレベル (level) に合わなかったため、再生が出来なかった場合に、サーバと通信を行い、映像再生装置のステータス (Status) を更新する例である。図25の終了 (2503) に代わって、図39では、ステータス (Status) 更新 (2503) が実行される (図39 (a))。図39の (b) に示すように、最初にアップデート・アプリケーションを起動する (250301)。このアップデート・アプリケーションは、プレーヤアプリケーションまたはゲームアプリケーション同様にミドルウェア上に築かれた一つのアプリケーションでもよく、あるいはオペレーティングシステム上で直接起動されるバイナリコードベースのアプリケーションであってもよい。例えば、ミドルウェア上のアプリケーションであった場合は、プレーヤアプリケーションが `Loader` クラスを介してアップデート・アプリケーションを起動することができる。

【0203】アップデート・アプリケーションは、ミドルウェア (Java (登録商標)) で提供される `Socket` クラスや直接ネットワークプロトコル (TCP/IP など) を使って、サーバと通信をする (250302)。通信するサーバは、`Package` の `<INTERNET URL="" />` によって指示されている。アプリケーションは、サーバと通信して、ステータス (Status) 更新をするために必要な条件 (金額) を入手し (250303)、ユーザに提示する (250304)。

【0204】アプリケーションは、ユーザからのレスポンスを待ち (250305)、ユーザがステータス (Status) の更新を望む場合 (250306) は、サーバとの課金処理 (250307) を行い、ステータス (Status) 更新処理 (250308) を行い、アプリケーションを終了し (250309)、プレーヤアプリケーションは図25のステップ2501からの処理を再び行う。

【0205】上記課金処理は、クレジットカード番号の入力と通信などによって行う。インターネット上での課金システムには様々な技術があり、また、本発明とは本質的に関係がないので、詳細な説明は省略する。

【0206】ステップ250306で、ユーザがステータス (Status) の更新を望まない場合は、そのまま終了する (250310)。

【0207】ここでは、ステータス (Status) の更新を例に説明をしたが、有効期限 (expire) の更新も同様に行うことができる。ただし、この場合は、映像再生装置のステータス (Status) 情報を更新するのではなく、パッケージの期限 (expire) 情報を更新する。パッケージが `ReWritable` メディアによって記録されている場合は、直接情報を更新することが可能である。一方読み取り専用メディアの場合は、映像再生装置が備えた不揮発性メモリまたはハードディスクなどの一時記録媒体に期限 (expire) 情報を一時的に記録するシステムを設けることで、更新情報を再利用することが可能である。

【0208】

【発明の効果】本発明の映像再生システムは、映画を単純に再生する映像再生装置としてだけでなく様々なアプリケーションを実現することを目的としている。このため、この映像再生装置では、内部メモリに予め読み込まれ実行されるソフトウェアとして、オペレーティングシステムの種類に対応した機能の差異を吸収するミドルウェアを含んでいる。このミドルウェアは、プレーヤアプリケーションが映像コンテンツを再生したり、ゲームアプリケーション等の拡張アプリケーションを実行するために用いられるツールを含むクラスライブラリを有している。具体的には、このミドルウェアは、上述のようにEパッケージ用のクラスライブラリを有している。ここでツールとは、各機能を実現するために用いられるクラス及びそのメンバ関数等である。また、パッケージメディアに含まれる再生制御情報 (管理情報) に記録されている機能リストには、プレーヤアプリケーションやゲームアプリケーションを含むアプリケーションにクラスライブラリによって提供される機能を記述している。また、この機能リストは各機能毎のステータス情報を有し、映像再生システム自身が有するステータス情報との比較によって、各映像再生システムで再生可能なコンテンツを機能レベルでコントロールすることが可能にな

る。

【0209】この結果、Eパッケージで実現する様々なアプリケーションを、ビジネスやサービスの形態や質に応じて制御することが可能になる。

【図面の簡単な説明】

【図1】 パッケージビジネスを示す概念図である。

【図2】 (a)～(c)は、コンテンツの価値を示す概念図である。

【図3】 映画におけるタイムシフトビジネスを示す概念図である。

【図4】 DVD規格の構成図である。

【図5】 家庭内におけるAV機器の構成図である。

【図6】 映画タイトル間でのリンクの概念図である。

【図7】 新たな価値を示す概念図である。

【図8】 Eパッケージのレベルを示す概念図である。

【図9】 (a)～(d)は、様々な規格の構成図である。

【図10】 ミドルウェア上でのプレーヤモデル構成図である。

【図11】 アプリケーションの例「プレーヤ」の概念図である。

【図12】 アプリケーションの例「ゲーム」の概念図である。

【図13】 アプリケーションの例「映画間リンク」の概念図である。

【図14】 Eパッケージ規格の構成図である。

【図15】 ディレクトリ、ファイルの構成図である。

【図16】 データ構造図「パッケージ情報」である。

【図17】 データ構造図「メニュー情報」である。

【図18】 データ構造図「タイトル情報」である。

【図19】 データ構造図「ストリーム情報」である。

【図20】 データ構造図「字幕ストリーム」である。

【図21】 ストリーム構造図である。

【図22】 映像再生装置構成図である。

【図23】 ソフトウェア構成図である。

【図24】 クラス一覧を示す図である。

【図25】 (a)～(c)は、Packageクラスの処理フローである。

【図26】 (a)～(d)は、Titleクラスの処理フローである。

【図27】 (a)～(e)は、Menuクラスの処理フローである。

【図28】 (a)～(c)は、Audioクラスの処理フローである。

【図29】 (a)～(d)は、EventクラスおよびLinkクラスの処理フローである。

【図30】 プレーヤ再生フローである。

【図31】 メニュー例の図である。

【図32】 タイトル再生中の動作例の図である。

【図33】 enableEvent関数の処理フロー

である。

【図34】 (a)～(d)は、Cursorクラスの処理フローである。

【図35】 (a)～(c)は、Statusクラスの処理フローである。

【図36】 (a)～(c)は、Canvasクラスの処理フローである。

【図37】 ゲームアプリケーション再生フローである。

【図38】 ステータス更新の概念図である。

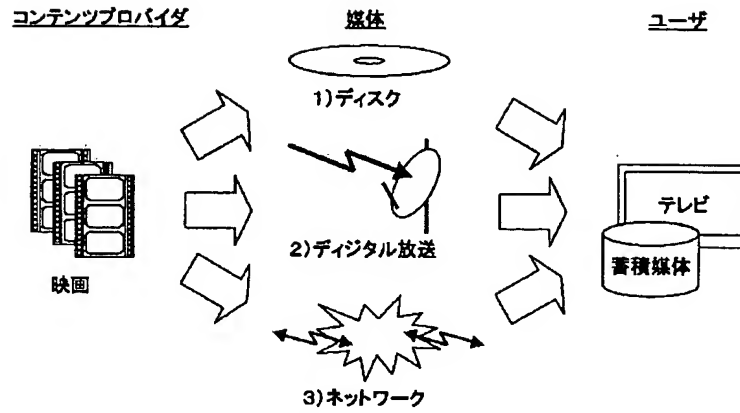
【図39】 (a)は、図25の(a)の部分図であり、(b)は、アップデートアプリケーション処理フローである。

【符号の説明】

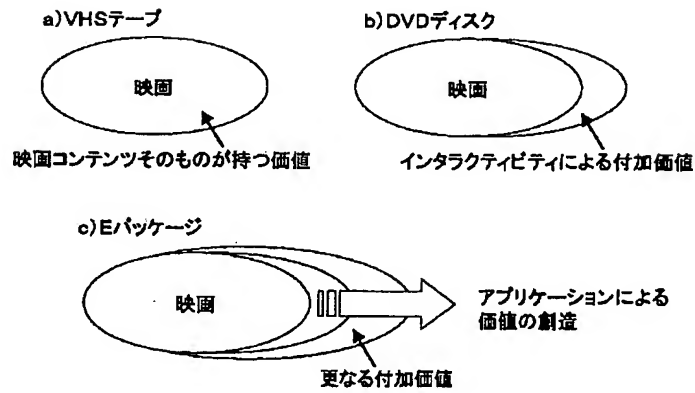
201 ファイルシステムドライバ
202 デバイスドライバ
203 オペレーティング・システム
204 ミドルウェア
205 Eパッケージ・クラス
206 プレーヤアプリケーション
207 外部アプリケーション
2501 package. XML読み込みステップ
2502 認証ステップ
2503 アップデート確認ステップ
2504 Menuインスタンス生成ステップ
2505 Titleインスタンス生成ステップ
2511 Menuインスタンスリターンステップ
2521 Titleインスタンスリターンステップ
2601 title. XML読み込みステップ
2602 Linkリスト生成およびLink先検出ステップ
2603 Chapterリスト生成ステップ
2604 stream. xml読み込みステップ
2605 AudioおよびSubtitleインスタンス生成ステップ
2606 Timelineリスト生成ステップ
2607 関数リスト生成ステップ
2608 Curosr生成ステップ
2611 有効確認ステップ
2612 再生開始ステップ
2621 Audioインスタンスリターンステップ
2631 オーディオストリーム設定ステップ
2641 スレッド起動ステップ
2642 対象時刻確認ステップ
2701 menu. xml読み込みステップ
2702 Titleインスタンス生成ステップ
2703 Pageインスタンス生成ステップ
2704 先頭Page表示ステップ
2705 menuThread起動ステップ
2711 次ページ表示ステップ

2721	前ページ表示ステップ	3314	execEvent起動ステップ
2731	タイトル通知ステップ	3401	位置情報生成ステップ
2741	スレッド起動ステップ	3402	通信開始ステップ
2742	Cursor通知受付ステップ	3403	CurosrThread起動ステップ
2743	Title選択ステップ	3411	スレッド起動ステップ
2744	selecteTitle起動ステップ	3412	移動確認ステップ
2745	Page移動確認ステップ	3413	moved起動ステップ
2746	次ページ移動確認ステップ	3414	選択実行確認ステップ
2747	nextPage起動ステップ	3415	selected起動ステップ
2748	prevPage起動ステップ	3421	位置情報更新ステップ
2801	stream.xml読み込みステップ	3431	Title通知ステップ
2802	属性値記録ステップ	3501	ステータス、期間情報生成ステップ
2811	言語情報リターンステップ	3511	ステータス情報応答ステップ
2821	圧縮情報リターンステップ	3521	期間情報応答ステップ
2901	Event属性設定ステップ	3601	Frame生成ステップ
2911	イベント処理ステップ	3602	デコード初期化ステップ
2921	Titleインスタンス生成ステップ	3603	オーバーレイ初期化ステップ
2931	イベント処理ステップ	3611	ストリーム情報読み出しステップ
3001	プレーヤ起動ステップ	3612	デコード設定ステップ
3002	画像生成ステップ	3613	デコード開始ステップ
3003	パッケージ選択ステップ	3614	描画開始ステップ
3004	パッケージ生成ステップ	3621	Frameサイズ更新ステップ
3005	メニュー取得ステップ	3622	表示サイズ更新ステップ
3006	メニュー表示ステップ	3701	ゲーム起動ステップ
3007	タイトル選択ステップ	3702	画面生成ステップ
3008	タイトル取得ステップ	3703	ゲーム開始ステップ
3009	タイトル再生開始ステップ	3704	パッケージ取得ステップ
3010	イベントスレッド起動ステップ	3705	タイトル取得ステップ
3011	イベント受付ステップ	3706	タイトル再生開始ステップ
3012	イベント処理ステップ	3707	イベントスレッド起動ステップ
3013	タイトル終了確認ステップ	3708	イベント受付ステップ
3014	プレーヤ終了ステップ	3709	イベント処理ステップ
3301	スレッド起動ステップ	3710	ゲーム終了ステップ
3302	時刻確認ステップ	3711	ゲーム終了ステップ
3303	ステータス確認ステップ	250301	更新アプリケーション起動ステップ
3304	分岐確認ステップ	250302	サーバ通信開始ステップ
3305	ユーザ要求確認ステップ	250303	ステータス更新条件入手ステップ
3306	タイムアウト確認ステップ	250304	ステータス更新条件提示ステップ
3307	Title生成ステップ	250305	ユーザ入力待ちステップ
3308	メッセージ確認ステップ	250306	ステータス更新確認ステップ
3309	Text生成ステップ	250307	課金処理ステップ
3310	Text表示ステップ	250308	ステータス更新ステップ
3311	タイムアウト確認ステップ	250309	アプリケーション終了ステップ
3312	Text消去ステップ	250310	アプリケーション終了ステップ
3313	イベント生成ステップ		

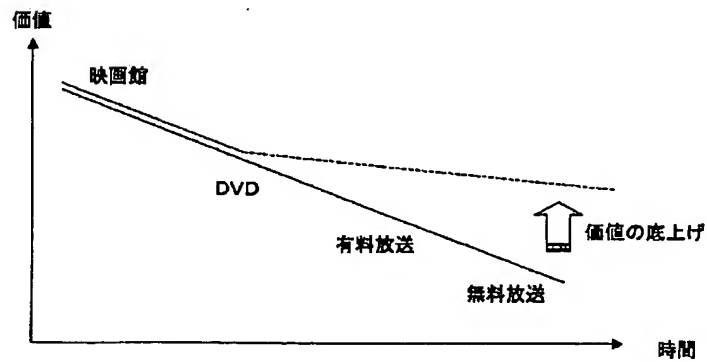
【図1】



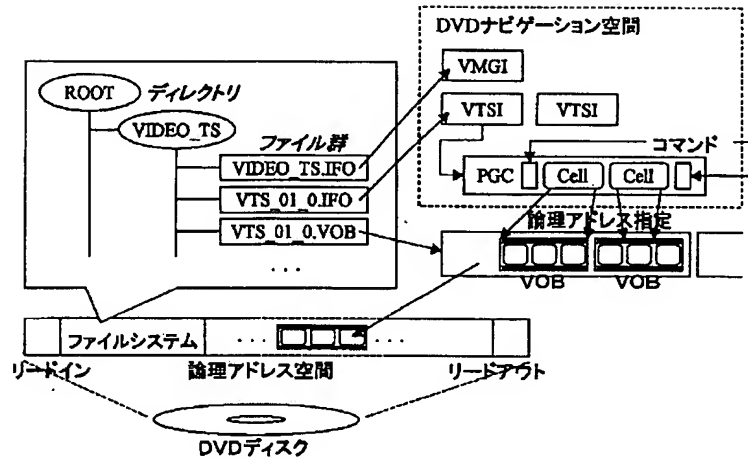
【図2】



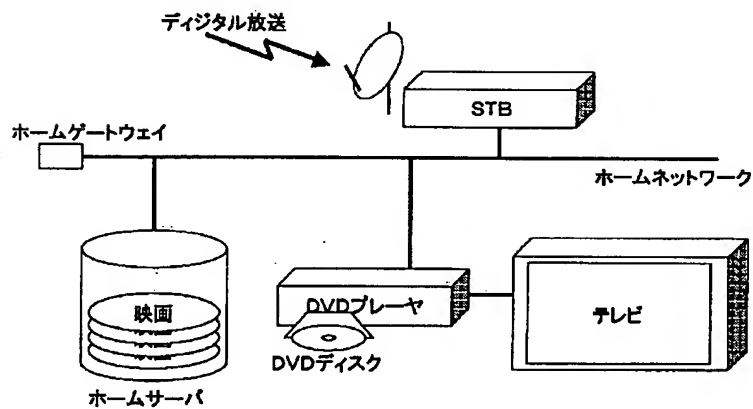
【図3】



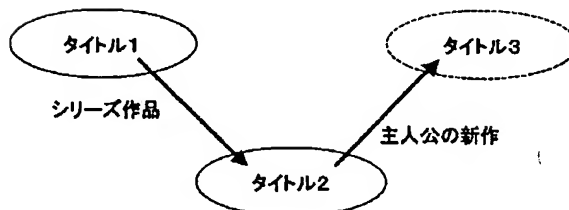
【図4】



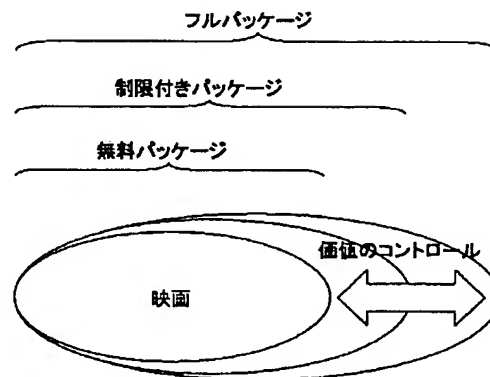
【図5】



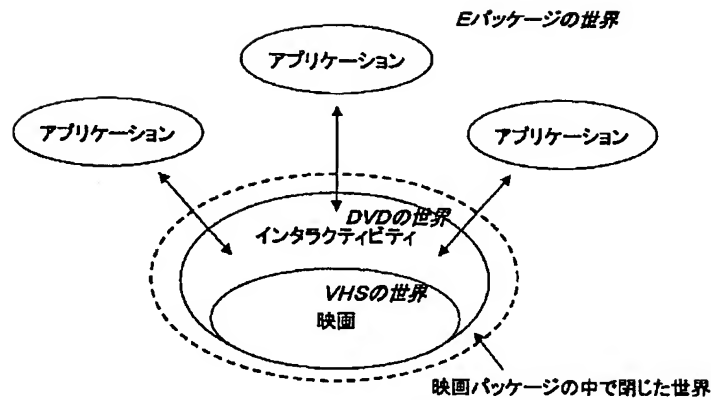
【図6】



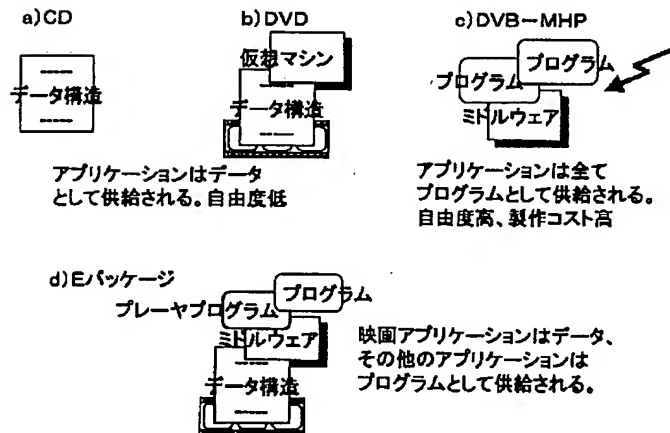
【図8】



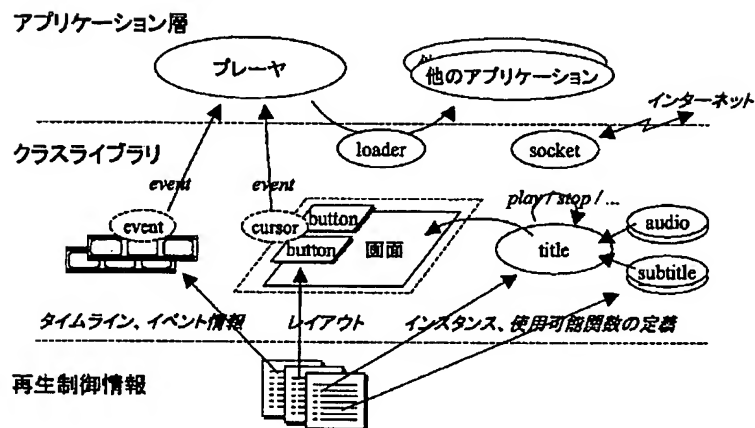
【図7】



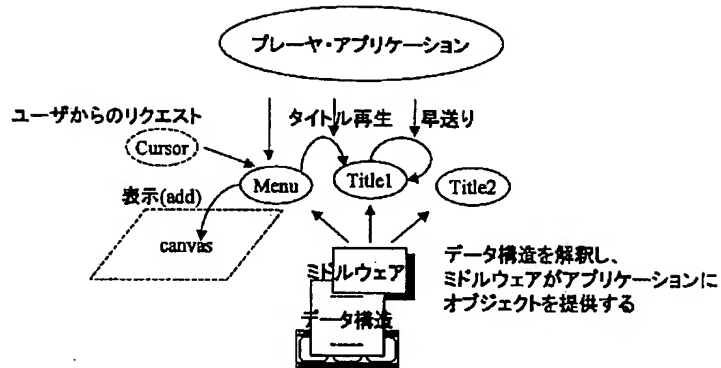
【図9】



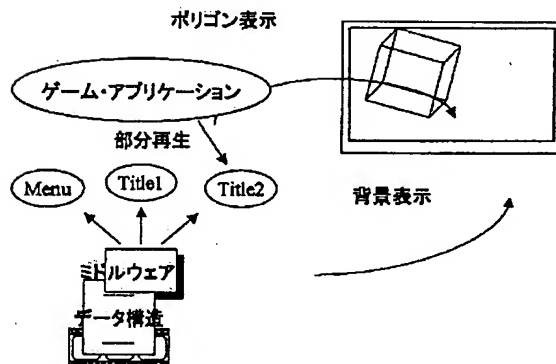
【図10】



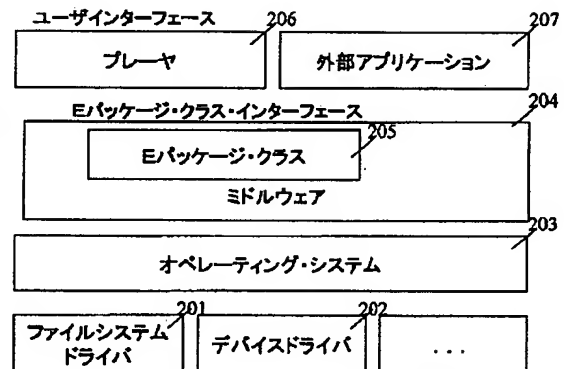
【図11】



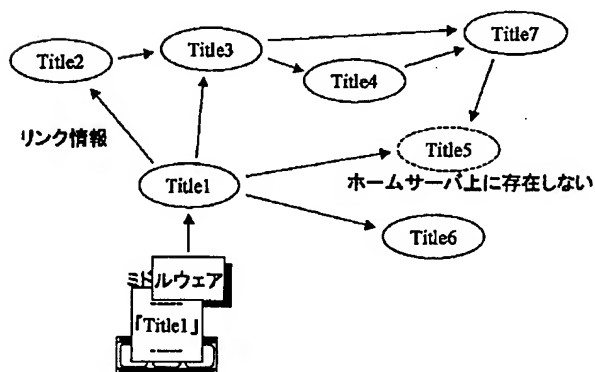
【図12】



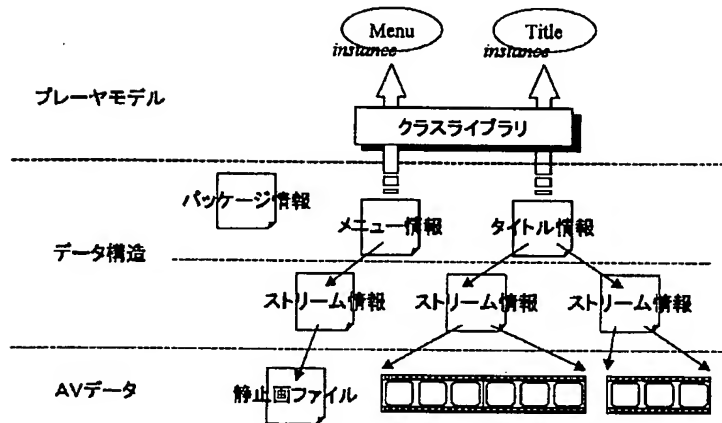
【図23】



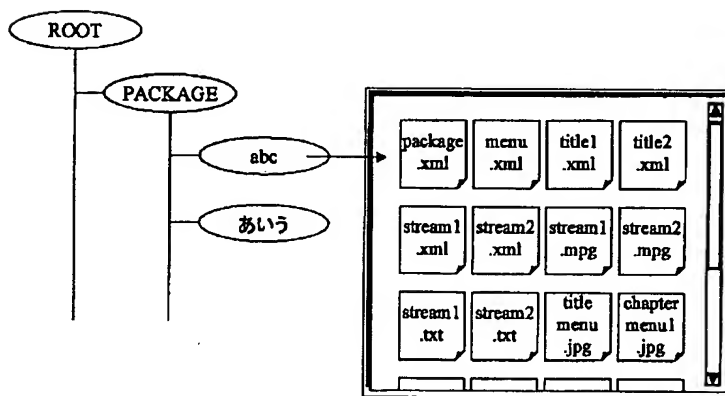
【図13】



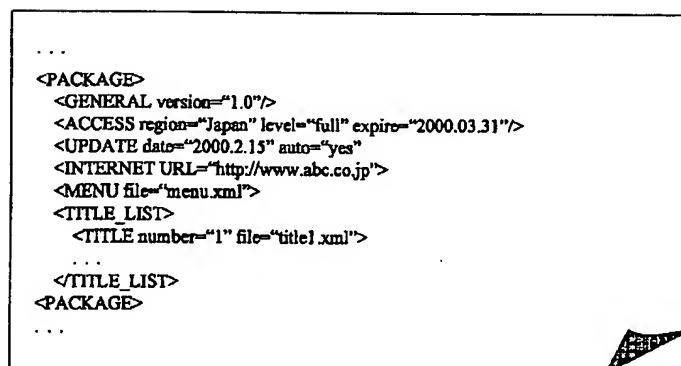
【図14】



【図15】



【図16】



【図17】

```

...
<MENU>
  <MENU_PAGE page="1" image="titlmenu.jpg">
    <TITLE column="1" row="1" title="1" object="button">劇場公開版</TITLE>
    <TITLE column="1" row="2" title="2" object="button">ディレクターズカット版
  </TITLE>
  </MENU_PAGE>
</MENU>
...

```

【図18】

```

...
<TITLE titleno="1" level="full">
  <LINK_LIST>
    <LINK ID="1" package="abc" title="1" chapter="1" time="00:00:00:00"/>
  </LINK_LIST>
  <CHAPTER_LIST>
    <CHAPTER in="00:00:00:00" out="00:24:15:30"
      video="stream1.mpg" subtitle="stream1.txt">
      <TIMELINE>
        <BRANCH level="full" message="過去の名作" id="1" in="00:01:00:00"
          out="00:11:00:00" jump="title2"/>
      </TIMELINE>
    </CHAPTER>
  </CHAPTER_LIST>
  <INTERFACE>
    <PLAY level="free">
    <SETRATE level="full">
  </INTERFACE>
</TITLE>

```

【図19】

```

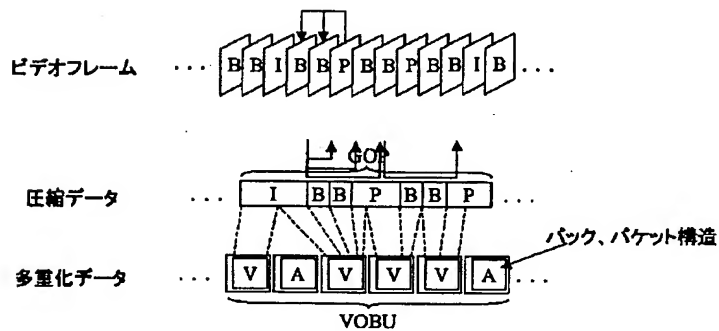
...
<STREAM file="stream1.mpg">
  <ATTRIBUTE>
    <VIDEO coding="MPEG2" resolution="720x480" aspect="16:9"/>
    <AUDIO coding="AC-3" bitrate="256kbps" channel="5.1" language="English"/>
    <AUDIO coding="MPEG" bitrate="224kbps" channel="2" language="Japanese"/>
  </ATTRIBUTE>
  <TIMEMAP>
    <ENTRY duration="00:00:00:12" size="152000"/>
    <ENTRY duration="00:00:00:15" size="213000"/>
  </TIMEMAP>
</STREAM>
...

```

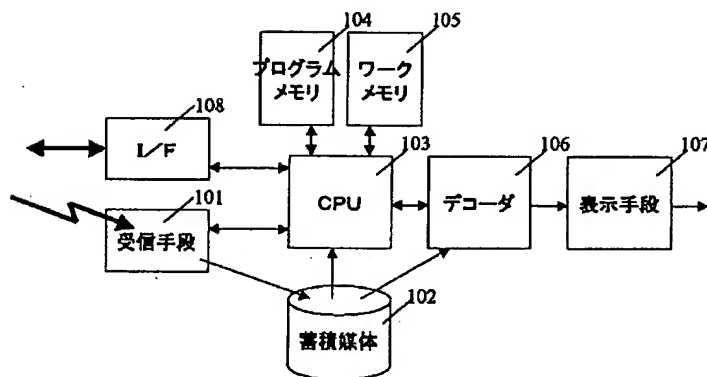
【図20】

```
...
<SUBTITLE>
  <LANGUAGE language="Japanese" character="Shift-JIS" font="Gothic"
    color="white" italic="off" bold="off" underline="off">
    <TEXT in="00:00:05:32" out="00:00:05:42">元気だったか？</TEXT>
    <TEXT in="00:00:05:45" out="00:00:05:50" italic="on">ああ、</TEXT>
    <TEXT in="00:00:06:02" out="00:00:06:12" color="blue">なにかあったのか？
    </TEXT>
    <TEXT in="00:00:06:17" out="00:00:06:25">じつは... </TEXT>
  ...
</LANGUAGE>
...
</SUBTITLE>
...
```

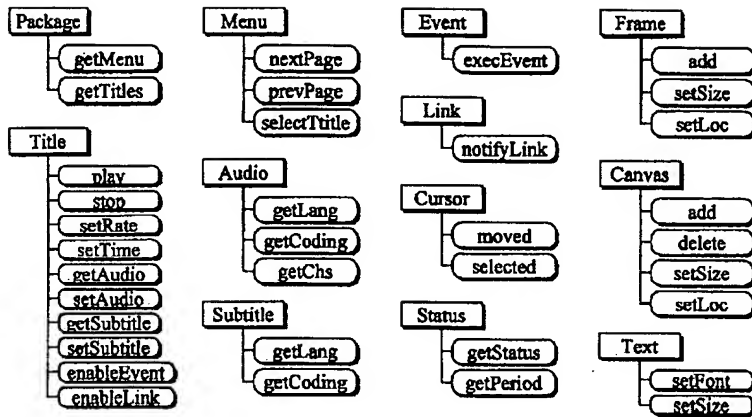
【図21】



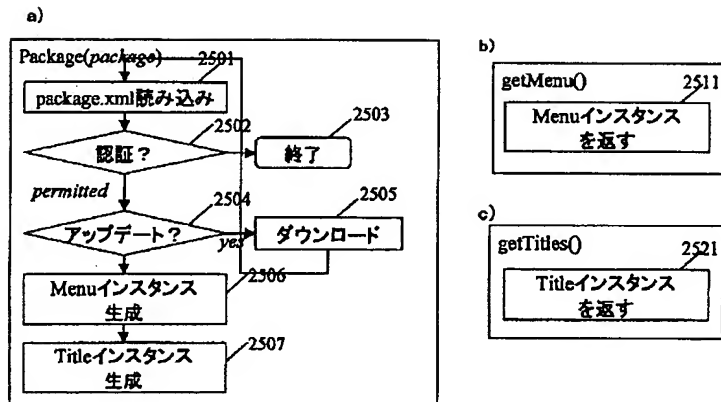
【図22】



【図24】

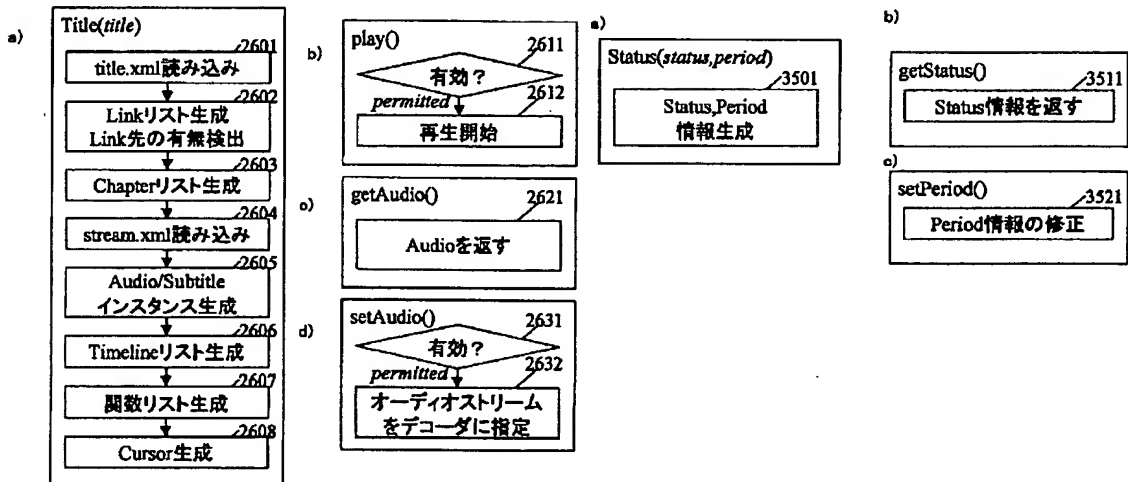


【図25】

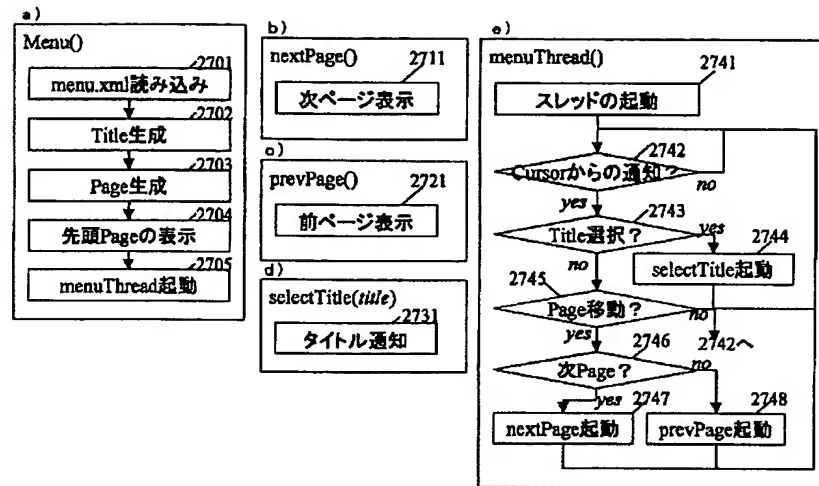


【図26】

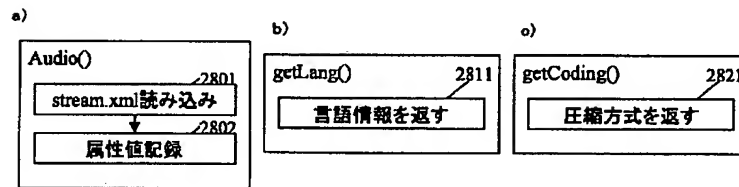
【図35】



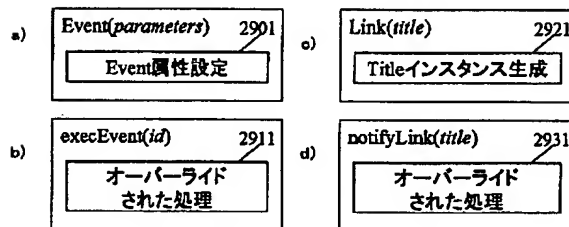
【図27】



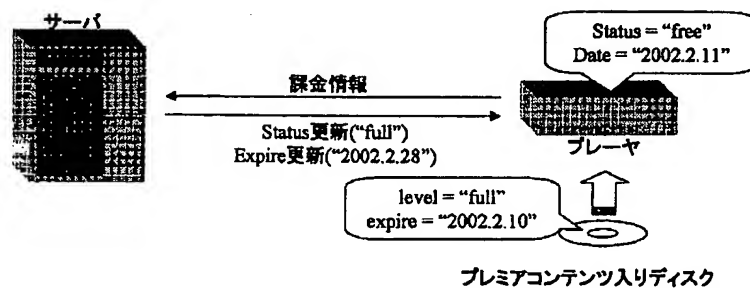
【図28】



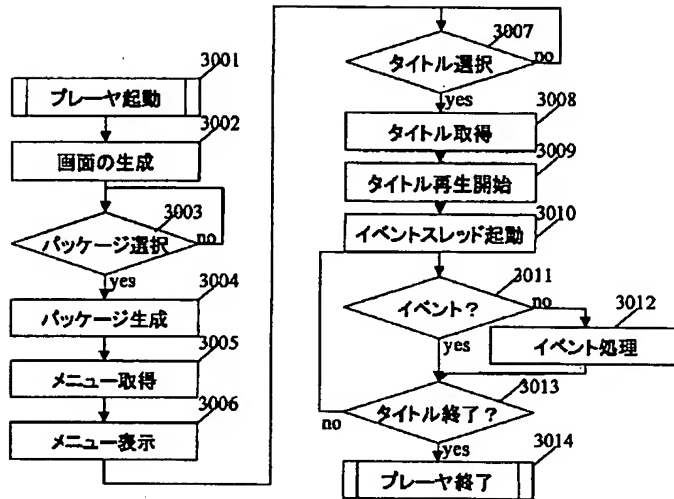
【図29】



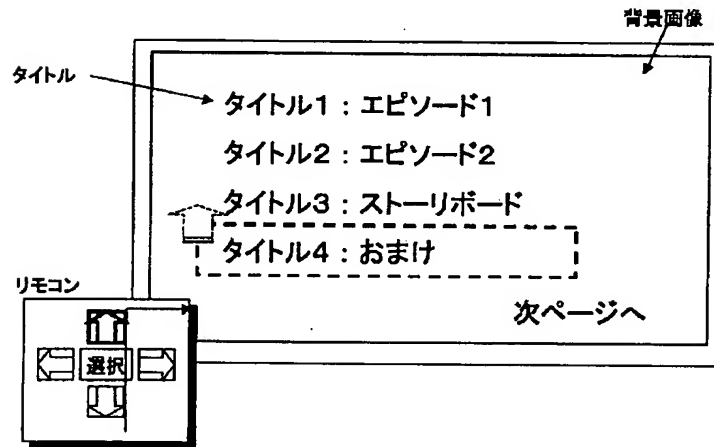
【図38】



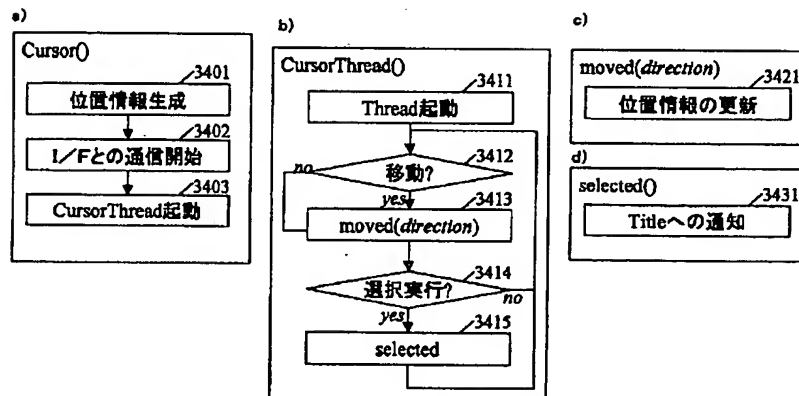
【図30】



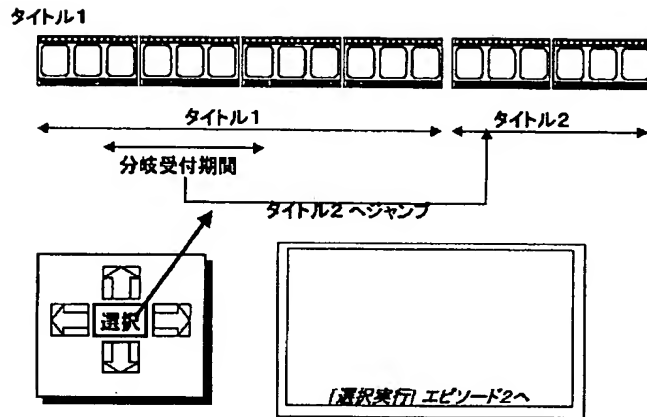
【図31】



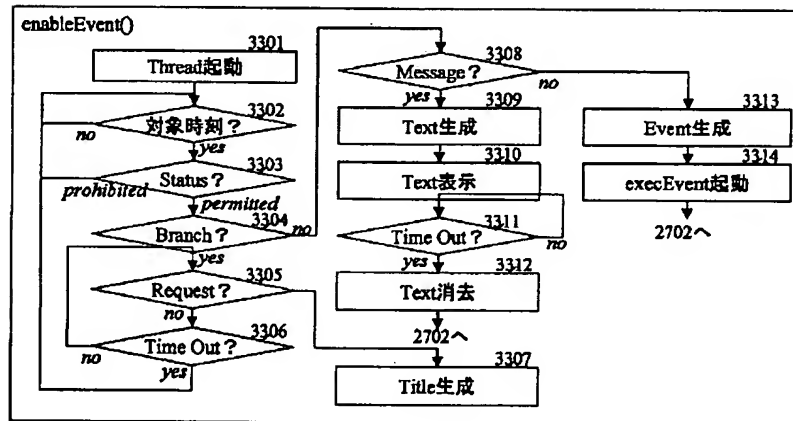
【図34】



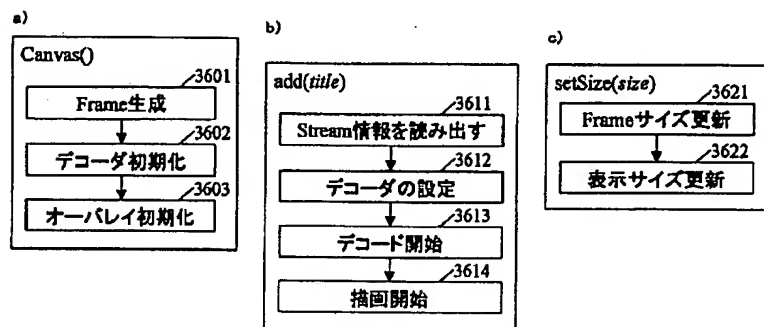
【図32】



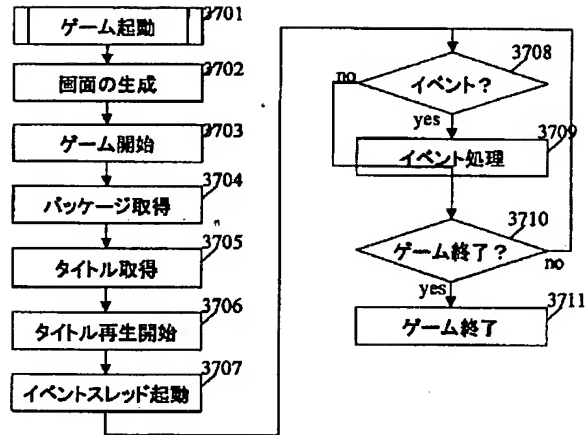
【図33】



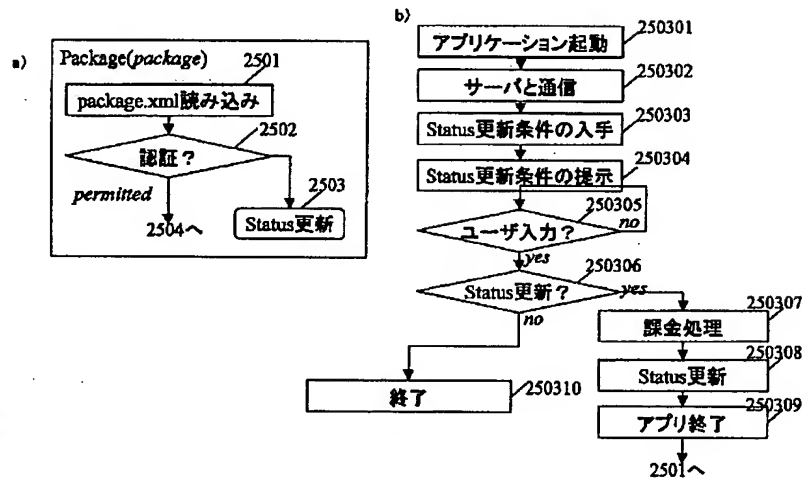
【図36】



【図37】



【図39】



フロントページの続き

(72)発明者 中村 和彦
大阪府門真市大字門真1006番地 松下電器
産業株式会社内

Fターム(参考) 5C052 AA01 AB03 AB04 AB05 CC06
DD04
5C053 FA23 GB06 LA06 LA11
5D044 AB05 AB07 BC03 CC06 FG10
FG18 FG21 GK08 GK12